An Efficient Information Retrieval System for Unstructured Data Using Inverted Index

[1]Abdullah Iftikhar, [2]Muhammad Irfan Khan, [3]Kulsoom Iftikhar

[1,2] GC University Faisalabad, Pakistan
[3]University of Agriculture Faisalabad, Pakistan

Abstract
The inverted index is combination of the keywords and posting lists associated for indexing of document. In modern age excessive use of technology has increased data volume at a very high rate. Big data is great concern of researchers. An efficient Document indexing in big data has become a major challenge for researchers. All organizations and web engines have limited number of resources such as space and storage which is very crucial in term of data management of information retrieval system. Information retrieval system need to very efficient. Inverted indexing technique is introduced in this research to minimize the delay in retrieval of data in information retrieval system. Inverted index is illustrated and then its issues are discussed and resolve by implementing the scalable inverted index. Then existing algorithm of inverted compared with the naïve inverted index. The Interval list of inverted indexes stores on primary storage except of auxiliary memory. In this research an efficient architecture of information retrieval system is proposed particularly for unstructured data which don't have a predefined structure format and data volume.

Keywords:
inverted index, big data, unstructured, auxiliary, algorithm

1. Introduction

The information technology and social media has a very vital role in present age. Now a days people wants to remain more up-to-date about what is happening around the world and what is latest in their interests fields. This has directly increased the data in search engines in exponential rate. On the daily basis data is generating in TB. This exponential growth in data is very serious challenge for the researchers who want to enhance the information retrieval. A data which is very large and complex to handle is called big data. It is increasing in exponential rate and its management is always remains a challenge for researchers. The inverted index is indexing technique of full text document. Inverted index is divided in front end and back end. The front end user directly searches keywords according to their needs using social media or web engine. The back end data is stored in some databases probably distributed database. That particular data is stored using some data structure on storage media such as primary or auxiliary media.

The inverted index is an index structure to map data files to its location, such as numbers and words in the database. It contains posting lists and term frequencies of document files tt are transfer, map and store in the database of computer. The data which has no predefined structure is called unstructured data. This unstructured data in information retrieval system generated voluminously, the information retrieval system which provide full text search to the users by using search engines are commonly comprised upon unstructured data.

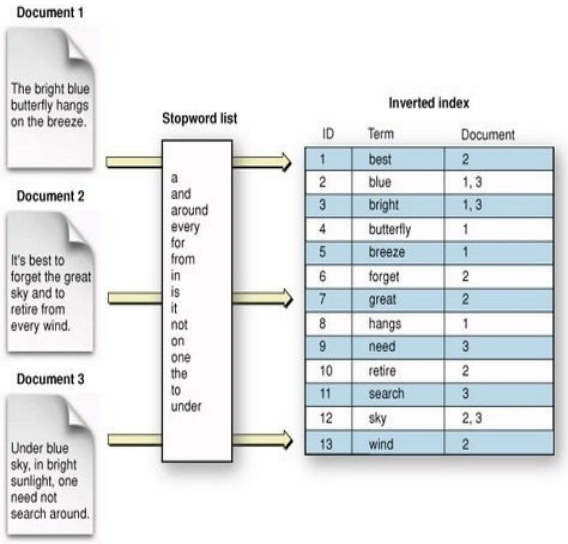

Figure 1: How to build Inverted index

The problem of which is mostly encountered in now a days data is available in a large size but it somehow do not match to every query or is amount of data that is not required by our query in information retrieval system. The fast and accurate retrieval is problem of this research. These delay may be belongs to the transfer of data in primary memory and virtual memory. The interest belongs to efficient and cost effective retrieval of data that is being searched using some search engines.

Manuscript received July 5, 2024
Manuscript revised July 20, 2024
https://doi.org/10.22937/IJCSNS.2024.24.7.4

The fast query result is also the demand of the research and data compression technique like run length coding algorithm contribute a lot to fast retrieval and data compression at the time of the coding and indexing in posting lists. Whenever the document is removed from database its index could track which consume crucial resources [1]. The appropriate information retrieval is more sensitive because it is not possible to done it manually to huge bulk of data. However, the information retrieval system deals spontaneously query search within indexed documents [2]. The document indexing is very critical process because system performance and speed of retrieval completely resides upon the structure and design of indexing technique [3]. The document relativity is examined via similarity of keyword and indexed document then this number of similarity is sorted in according to module paradigm. The information retrieval system is about management, organization, and sharing the information entertaining to the given queries. It is not always possible to the retrieval system to return useful and meaningful data it is just the pattern matching similarity process which describes the prevailing documents according to query through search engines [4]. Keywords are used to find out query from document, this query is given some weight according to system specifications that refer high weight to high priority and high similarity then this weight is included to the inverted index file or posting list and use to notify the result of the query [5]. Document requires indexing technique to search query from database. Inverted indexing is well defined structure of data storing and retrieving from provided dataset. It consists on inverted list, inverted files having number and frequencies referring to the dataset. The performance of information retrieval system could be enhanced through the analysis and compression of data that is in raw material and need to generate useful information for query and inverted index optimization [6]. The frequencies of data in document term should scan first before constructing the inverted index of the document [7]. The benefit of that scanning is that system will defined the finite state machine to parse the term document will then minimizes the regular expression of that grammar [8]. The image based information retrieval system is discussed as the color of the image shape and texture. Color management and storage of the image is easily manage and accessible but shortcoming is that it completely neglects the texture of the image at the time of the precision and the recall [9]. The region based Image segmentation algorithm firstly introduced but later on it finds as an element which slows down the retrieval of image from information retrieval application during precision. There are many techniques to segmented the image histogram is the most commonly used technique to convert image at a grey scale for better resolution of pixels and image [10].

2. Inverted Index

Inverted list is a combination of unique identifier of frequently appearing strings of inverted indices whether it is post or pre-string in the document. Inverted lists consist of term appearance in the document and location of the particular document on disk [11].It may be extending in term of identifier length. Another term is used in inverted indices is a query resolution that use to fetch and decode query for the inverted lists. It also checks the query relativity among the document and computes it for the inverted lists. It presented query answer that term which has high relativity in the document of inverted list to the inverted indice [12]. Map reduce is a modern programming model which is best suitable to handle large amount of data (big data). It is suggested to speed up indices construction using its modern frameworks and algorithms to manage all kind of data and required management technique [13]. In memory inversion, sort based inversion and merge based inversion are major three types of building inverted indices in traditional stand-alone environment. Inverted indices is a recommended for search engines for efficient retrieval of information retrieval system [14].

Inverted indices somehow referred to as a ranked query as it take into account the number of occurrences in descending order.

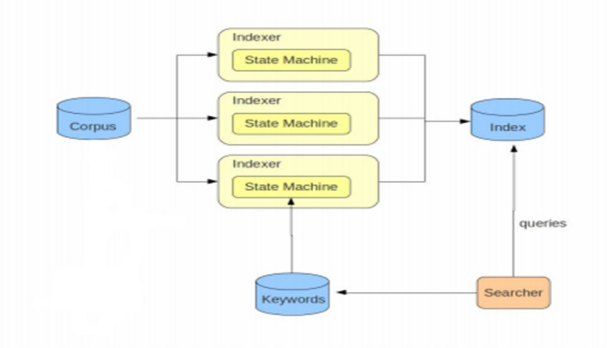

Figure 2: Structure of Inverted Index

The above mentioned figure is depiction of the inverted index parallel query processing paradigm where all finite state of automation works independent of each other. Inverted index is used in combination with corpus technique of indexing for maximization accuracy and actual data retrieval. Data is crawl from back-end to forefront, user parse his queries as keywords. it is an important information to kept in view that indexing in not at all free of cost in inverted indices as you want to compress a document in inverted index it almost occupy same space as real text occupy in text it is crucial that compression of data should betake place at the inverted lists it considerably decrease the

space. Researchers always remained concerned about the compression of inverted lists. The classification of document gaps is a simple method of compression through deduction of inverted lists. Frequent term has less document gaps while rare terms has longer gaps .gaps are referred to as differences in the consecutive numbers of term occurrences in ascending orders.

Modern search engines rely on indexing techniques centralized and distributed indexing are main distribution of indexing techniques [15]. As data is growing exponentially that's why researchers consider distributed technique more reliable and efficient as it could easily handle large amount of data which is generated by all major search engines for indexing [16]. Big data is increasing at enormous scale quick retrieval of large space required data which is a challenge because analysis and fast retrieval is not possible such as pointer links of inverted indices are stored on auxiliary storage [17]. Centralized databases provides magnificent support to structured data but to deal with unstructured data is not an easy task somehow of velocity, variety and volume. Inverted index is an indexing structure support by almost all search engines [18]. When result is presented to a user top ranked or high priority is shows to the user. It is important to take into an account that priority of document also subsiding function of the term frequency and length of the term inversely related occurrence of particular term in the text.

3. Unstructured data types

Unstructured data is most probably characterize by the highly presence of the digital content, Text based enormous dimensions (big data) of content [19]. In present age data is generated at a very high rate which has made it more complex for search engine to analysis of data in high quality and relatively in short period of time [20].

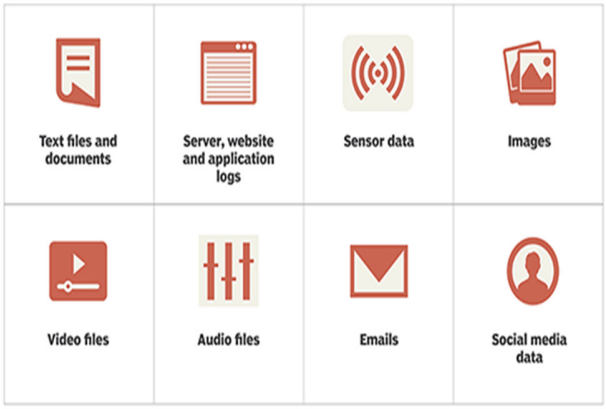

Figure 3: Types of unstructured data

Above mentioned includes the commonly used types of the unstructured data. Unstructured data containing is explained as:

3.1 Text files and documents

The text file refers to computer files that do not have particular format and image information; it is just line by line classification of words. These files are recognized by the extension at the last with .text files [21]. The text files are container of text content in computer system. There are two categories of computer system files text files and binary files to carry electronic text. This data do not have predefined structure for architecture of computer system.

3.2 Servers and website

The servers of any system hosts web or nodes of that system and provide facilities to the entire interconnected system and allocate resources of crucial need in efficient manners. All-important files that need to be share, maximize, secure and communicate for system, are available on servers of that system.

Website or site is collection of webpages hosts by some servers in some fashion of groups. The most of un-structured data is generated by the excessive use of the websites [22]. It generates data in large bulk as it is combination of webpages that it contains documents which is access by using web browser.

3.3 Sensor data

In the age of information technology sensor data is most of type generated data, as we consider it as data that is outcome of the electronic devices connected to the computer input some data by system and generate output to that input is called sensor data [23]. These devices are commonly used now days for different causes such as security, household, forecast, sensors for diagnosing diseases etc. This data output is mostly used as an input for any other system. It is the type of data, that sensors has completely control and manage human life as it resides more on electronic devices.

3.4 Images and emails

Images and emails are widely used now days. A very large amount of unstructured data is generated by the excessive use of them. An image is pictorial representation and storage of physical objects into electronic form in any computer system. The electronic mail usually called email is sent alike letter, from one system to another system in electronic form travelling through internet.

3.5 Audio and video files

The audio and video files format do not specify the structures of files hat why it needs more intense information about file storage and indexing for efficient retrieval information while passing query as keywords to the database. These files may be compresses and may not be. Compression of audio and especially video files reduces size mostly using lossy compression technique. Video files are heavy to files system, it needs maximum resources of the system there compression is necessary to overcome resources utilization at their behalf. There are many techniques and algorithm available for compression but it has drawback to slow down the speed of the system.

3.6 Social media data

The excessive use of information technology lends social media excessive use. Now, more and more people are using social media sites to stay connected to each other but also remain to keep in touch about what are new updates about world. What is new and latest trends generate huge amount of data generated by single user. Facebook, twitter, instream, are most common sites to generate unstructured data. These sites are great source of entertainment also as there is almost all information about particular interest.

4. Materials and Methods

Big databases manipulate billions of documents containing thousands of queries within second. Efficient and fast query retrieval is a major concern of researcher. Inverted list could easily expand from MBs to required GBs which is a bottleneck of performance enhancement. The Increasing trend of technology leads the way to depend upon search engines to access information [25]. Compact data structure combination with treap topology help in compression of document in given dataset [24]. Internet access is increasing day by day as a result it produces many challenges for researchers such as data organization and searching operations on that data in term of efficient retrieval of information. Space consumption is also keeping researcher engage to investigate new techniques for data indexing [26]. Text retrieval is essentially to care of because a lot of communication convey through it [27]. Inverted indices are a look up table which consist of substrings of textual data of document databases. Inverted indices are a look up table which consist of substrings of textual data of document databases [28]. Inverted indexes is a collection of related code words defined in some codebook where clustering is impose on primary data of vectors [29]. The main aim of inverted indices remains to bring query vector adjacent to database vector. Codes words are sorted in descend order and list of few code words combined to response query request [30]. It always excited researchers to reduce the distance between query and dataset as it is not stored on internal memory however on external memory [31]. Inverted indices never required primary dataset to process query. It required interval list to answer any query [32]. It was stated that during some decade inverted indices and bag-of-words remain only source of image retrieval in large databases. Inverted indices were single minded firm to retrieve image through using its discriminative ability [33].

It was stated that inverted indices or files are vital important in information retrieval system to search particular material mostly text data in unstructured nature data [34]. They proposed to build a system which main aim is to provide authentic information relaying on user requirements, individualities and behavior for thematic virtual museum system [35]. They provide relevant and useful information and do not puzzle user by providing irrelevant information. They needed to collect document based data collection for optimal retrieval and suggested construction of thematic virtual museum inverted indices. (IDF) Inverted document frequency and (TF) term frequency were techniques introduced to investigate status of information collection which were evaluated in descending order [36].

The information retrieval system was completely resided upon the data generation sources such as search engine. These search engines generated data that was not easy to distinguish its nature and structure [37]. The information retrieval system was replaced by the data-spaces that worked in corporation existence technique [38]. Their main feature was the integration less structure to search a query into the given database. They introduced data space support platform to search query from database by using the principle that

1.It was especially designed to treat the heterogeneous databases.
2.The data space support platform did not provide the full control on the databases such as transaction and indexing it was just designed to deal with data-space component services. The other integration system was controlled by the data management system.

3. The data-space source platform was very efficient for query optimization process. When the user enter query using keywords for single unavailable data then it returned optimize query by making search collective data search.

4. The data-space support platform was designed to compensate the space resources by providing them mutual exclusiveness.

Investigated that with the development of web, clients were looking forward for reports that were only

available on request. The development in data volume influences the execution of retrieval frameworks [39]. In any information retrieval framework, two fundamental variables considered were reported representation named as indexing and retrieval [40].

5. Information retrieval system

The information retrieval system referred towards the finding information mostly of unstructured nature (documents) from the computer connected with internet [41]. The architecture of the information retrieval system categorized in two parts, indexing and searching documents [42]. The text retrieval efficiency is the need of the time as generally it provides the public access to internet connected libraries that more and more people are searching their required document and information demanded the efficiency and the accuracy of the searched data

5.1 The design of inverted index

In the present age standard methods of the indexing sorts are, inverted index, index count what more signature innovation. index postfix cluster methods remain great by state question; innovation of mark record utilized reduced right now; Inverted indices additionally recognized as an indices, inverted situation records, what more turn around the document. Inverted indices are an index structures which utilized catchwords to index the documents. It is very much found develop variation of data retrieval techniques, giving efficient retrieval of information. The storing structure of information retrieval proficiency then adequacy of retrieval system plays vital part and apart from that it also enhanced retrieval designs. There are two sorts of primary storing type of inverted index

5.1.1 The records representation in inverted index

It keeps record of each referenced word of the text within located file. Just consider the English language as an example, listed below content needs to be indexed .the T0, T1 and T2 are term to use. Where T0="it is the thing that it is"; T1="what is it"; T2="it is object". we supposed the additional inverted indices record. "a":{2}; "object":{2}; "is":{0,1,2}; "it":{0,1,2}; "what":{0,1} the record of given data file or text keep in record ever single word shows up the required report, however the exact location where each word in the document is located do not acknowledged to the user or customer . consider that the client is looking towards condition "what is it" there are possibilities that it may considered it as the "what", "is" else or "it" then correlate with above mentioned data values set: {0,1}∩ {0,1,2} ∩ {0,1,2}={0,1} showing as it is the part of given set T0 and T1, however the exact response related to the content of T1 set values. It is the redundancy that is ambiguous to the client that either query response to which exact data set duplication of record is the reason of the ambiguity.

5.1.2 Word level inverted index

The word level inverted index construction is the prominent as just not only stored the word in the databases but also keep in record the exact location of that particular text files documents. Its shape structure gives greater similarity, (for example, express hunt), however requires additional time and space to make. On the off chance that regardless we utilize three messages. "a":{(2,2)}; "object":{(2,3)}; "is":{(0,1), (0,4), (1,1), (2,1)}; "it":{(0,0), (0,3), (1,2), (2,0)}; "what":{(0,2), (1,0)} as mentioned before in the documents it, not only specify the word name but also provides the complete information about the exact location of records in the data set(database). For example, the "object": {(2,3)}, demonstrated that the term object is in the T2 located at the fourth word of data set (starting addressing from the 0). At the word level indexing using inverted index the required term for example "what is it", will be clearly leads towards the T0 or T1. Notwithstanding, it enhanced indexing designs to find out the correct and exact to discover "what is it" just in the T1. The inverted index is referred to as the collection of keywords in the documents having similarity on the bases of the term frequency assigned to it. The inverted index is combination of the all the index entries with in the single index file while a single file in the documents term also take a complete list of similar frequencies . Whenever a new file is to be inserted in the inverted index the whole paradigm is re-indexed which becomes the very much serious issue to the indexing when library of documents is publically accessed means randomly changed.

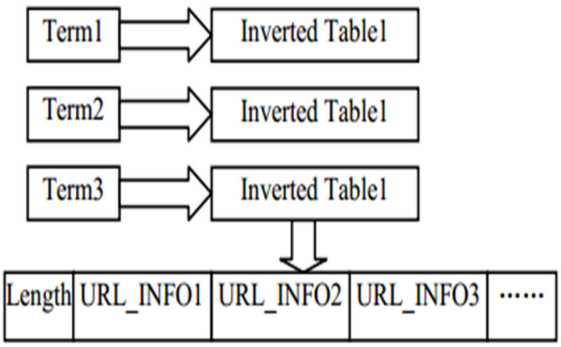

Figure4: The inverted indices table structure

This is hard to achieve the indices advancement specification overhead current document framework, at the point when the multi-words question, it would make prohibit and intense decrease in execution of query being indexed and searched [43]. So as to survive the shortages of

Inverted indices total address inverted indices innovation paradigm is suggested to tackle the shortcoming.

5.2 Technical analysis

The inverted indices revolution along with the compression revolution lessened has volume of inverted indices of outer storage devices, it enhanced reversal table structure, diminished substance which entrance is prohibited reversal table circle; the re-positing the indices framework sensibly diminishing seasons of Input and output [44]. The inverted indices revolution could successfully accomplished decided quantity back to back watchwords; this utilizes the repeated words for the productive structure of indexes [45]. As simple as that the records library is similar to document file and the position of that particular in the database for retrieval of exact and required information in below table.

Doc_Term	File-Name	Start-Pos
Term_1	Filename_1	Startpos_1
Term_2	Filename_2	Startpos_2
……..	…….	…….
Term n	filename n	Startpos n

Table 1: Record library table

The other part contains reversal table known through the document name, moreover, fundamental thought watchwords comparing the inverted record, appeared in second table. The Inverted record as indicated by a specific system store the index in the framework circle as a characterization, and the framework utilizes the dispersed capacity methodology, putting away index arrangement is in distinctive hubs, you can rapidly look through the required indices information smaller measure. It is suggested for specific degree to which proposed measure decreases issue of input and output over-burden because of the reversal table as well colossal.

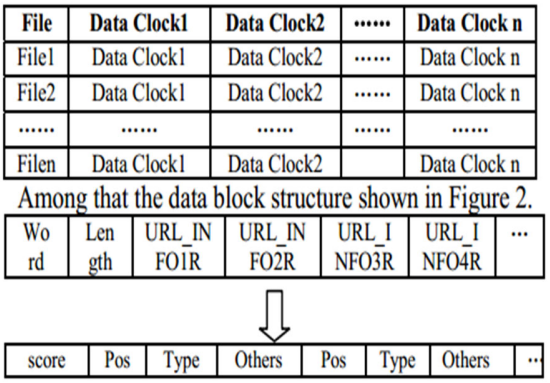

Figure 5: The data blocks structure in inverted index

5.3 Query design implementation

Inverted index maximize utilization of address inverted index revolution, incorporating single and multiple words search, utilizes consistent multiple words design coordinating innovation, it must take care issue of the seeking and arranging of the word what more multiple words, and coming to strong relationship.

5.4 Relevance ranking

At the point when all the inquiry watchwords have wrapped up the indices database, it utilized a progression based upon website page. At the point entering a catchphrase about to search, it is supposed to be a progressively watchwords coordinating all databases, more drawn out the coordinated watchwords are, additionally going before the outcome[46]. It involved variety of the inter-fix of website weight, and also kept in record the higher ranked web page, then inside and out investigation of the examination of various tests, summed up the accompanying technique for ascertaining weights, tests demonstrate this can guarantee a sufficiently high relationship without a misuse of registering assets. In the tremendous information of web page, it inherit the web page and higher weight recorded as a parent web page for document indexing, it is kept sorted with the exact location pointers. That is, a page is called parent page of in the circle of indexing others pages communicate to successor web page. Obviously, all ancestors and successors web pages in the circle becomes parent web page while being child web page of another web page. In the event that C, D, E, F all connected An, at that point the announcement A web page additionally is the most essential, the web page is additionally the most astounding frequent esteem. In this way, the more which is cited, the more which will expand the heaviness of the opposite, in actuality, the more which cites, the more which will decrease the weight. Along these lines, previously returning arranging, first we utilize Page

rank calculation to sift through the portion of the less frequent pertinent web pages documents.

PR(A)=(1-d）+d(PR(T1)/C(T1)+… +PR(Tn)/C(Tn)

Among it, PR(A) is a web level of page A; PR(T1) is a web level of page T1, page T1 is interconnected to A, C(T1) referred the total connections build on the parent web page. Page-rank calculation computes the nature of site connected web pages to the system arrangements for sorted website pages [47]. Computing real circumstances a page examined by reference number of documents, a site web page estimated the level of consequence in the totality what's more, in this manner as indicated by Page rank procedure arranged by the principal of arranged for web pages, which belongs somehow to degree, have the capacity to sift through blunder due to malignant word recurrence what more the location of word.

$$v=\sum 10n+\sum(10\times k2)+\sum Dis()$$

The n referred towards the total numbers of watchwords used or stored in the document file, also called word recurrence. Word recurrence referred to the total number of sorted words of a database file. When any word of the document files being query at the multiple webpage at the same time it automatically strong the words frequency and it is ranked to higher query response fields of catchphrases. The term "k" is the arrangement length of continuous current coordinating watchwords; D is () is the capacities out there frame n to n-1 words, you can get:

$$D \text{ is } () = 500/ [(Xn-Xn-1)*(Xn-Xn-1)].$$

5.5 Structure of inverted index

We define inverted index as a collection of keywords and posting lists related to their stored documents [48]. Posting lists are defined as an individual having unique document id (doc-id) and exact location of that document[49]. Inverted index could also explain by the term frequency which indicates the presence of particular term as mentioned before by the reference pointer. The scalable and efficient algorithms for documents and indexed construction is defined within the available resources of hardware keeping in mind the availability of that resources to the common man block sort indices and standalone pass in memory are algorithm that are supportable for a common man. The description of term and their postings frequency is mentioned below:

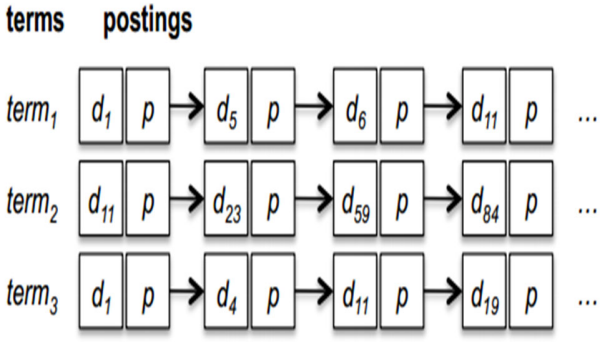

Figure 6: Posting list with unique id and frequency

The above mention figure explains very briefly about the presence of the text which is referenced to the posting list and each postings list unique id of document and term frequency [50].

Mappers emit postings keyed by terms, the execution framework groups postings by term, and the reducers write postings lists to disk.

```
1: class Mapper
2:     procedure Map(docid n, doc d)
3:         H ← new AssociativeArray
4:         for all term t ∈ doc d do
5:             H{t} ← H{t} + 1
6:         for all term t ∈ H do
7:             Emit(term t, posting ⟨n, H{t}⟩)

1: class Reducer
2:     procedure Reduce(term t, postings [⟨n1, f1⟩, ⟨n2, f2⟩ …])
3:         P ← new List
4:         for all posting ⟨a, f⟩ ∈ postings [⟨n1, f1⟩, ⟨n2, f2⟩ …] do
5:             Append(P, ⟨a, f⟩)
6:         Sort(P)
7:         Emit(term t, postings P)
```

```
1: class REDUCER
2:     procedure REDUCE(term t, postings [⟨n1, f1⟩, ⟨n2, f2⟩ ...])
3:         P ← new LIST
4:         for all posting ⟨a, f⟩ ∈ postings [⟨n1, f1⟩, ⟨n2, f2⟩ ...] do
5:             APPEND(P, ⟨a, f⟩)
6:         SORT(P)
7:         EMIT(term t, postings P)
```

The above mentioned is the algorithm of the inverted index in baseline case. For this kind of algorithm map reduce was designed to tackle the issue of more storage consumption and less accurate data what more particularly the small file document paradigm. It is very fundamental algorithm of indexing of documents files. It is very obvious that the documents are processed parallel in the mapper paradigm of algorithm. Then it simply uses similar keywords tokenization and eliminates all stop words from the documents and makes it more similar to understand by the indices, the next task is all about the elimination of the affix such as from "cats" to "cat". When all preprocessing operation are done to the documents the very next step is all about the building of the term frequency which keep in record the same term appearing in the database along with its count number od given word. The above mentioned pseudo code kept in secret the actual scenario of the execution of actual processing in the data somewhat explains it in the 4 to 5 line.

The very next in pseudo code mapper repeat the term operations by term unique id (Doc id) and assigns term frequency to values. The [n,H{t}] explained about the postings of the doc explained by a demonstration figure as above. The 5 and 7 lines are all about the mapper assigned key value and the pointer postings of the data according to the mapper pseudo code.

The later it becomes obvious that the mapper emits all values assigned to it and just only stores the term frequency of postings document that's why it could easily expanded to store about the additional information of the data. Map reducer here explained by the distributed term utilization in the data term frequency. The researcher and developers do not need to perform any action to bring about all participating posting lists. It is very helpful for the reducer paradigm in the pseudo code part such as it does not need to recall individually all participating posting to write them on hard disk (storage medium). The reducer is responsible for all about the initiating the vacant list then all corresponding responds to it having common key values within. For postings list efficiency it is very necessary to compress their records for optimized query result within time and accuracy. Finally the residing key values are stored and indexed according to the inverted index structure and implementation.

The inverted indices size could easily expand according to the term frequency stored in it. It is very important to consider that the inverted indices in the optimized scenario and paradigm are only the 1/10 of the actual indexed data in the data set. It is not the same case when we are talking about the reference list storage or pointer storage it becomes exceptionally larger and consume auxiliary memory.

5.6 Algorithm execution

The below mentioned diagram is the complete depiction of the above mentioned inverted index pseudo code participating 3 mappers, 2 reducers what more, 3 documents. It is seen very clearly that the mapper key value pair and reducer's final key value pairs participating in the inverted index paradigm are demonstrated by the dot lines. The term frequencies are at the right side while document containing unique id is at the left side of the document. Postings demonstrated box pair in the figure. The map reduce is the very accurate and brief depiction of the inverted index paradigm in the given example. If the document database is not too large then few lines of Hadoop paradigm are required to gain efficient and accurate results. The map reduce facilitate the programmer in such a way he should not bother about the required space either low or crossing to the limits regarding documents and its indexing. The following diagram is very simple and easy depiction of the map reduces in the inverted index paradigm.

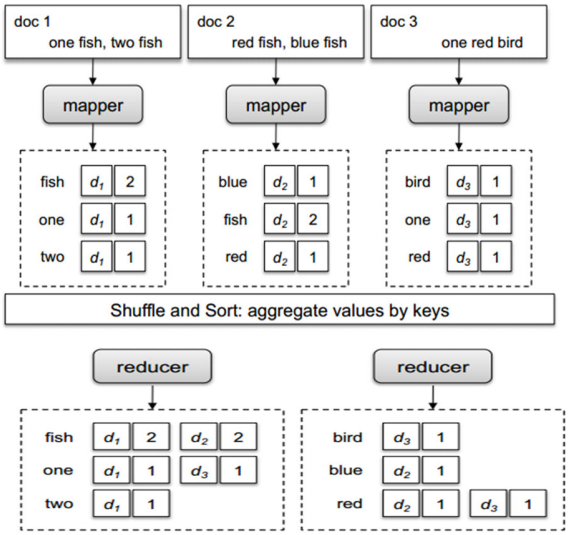

Figure 7: map reduces in the inverted index paradigm

5.7 Reviewed implementation of inverted indices

The previous mentioned algorithm of inverted indices somewhat provides baseline structure facilities but at the same time it does not show much scalability as the dynamically ration of inverted construction grows in order. If we look upon the algorithm then it becomes obvious that there is the large scale memory utilization storage guarantee provided by the algorithm pseudo code paradigm but in the ground realities it is no more scalable and does not also guarantees that the postings data will be accurate and order of that data will be consistent in the memory it is indexed at the initial stage.

This problem of order and storage could tackle with the help of the surety that reducers only receives id and key values pair in the sorted order. The map reducer should only focus on the sorting order of key values pairs except of the eliminating affix such as “cats” to “cat” in the document. Such as:

$$(\text{term } t, \text{posting } \langle docid, f \rangle)$$

It is recommended that except of single key value pair elimination it must eliminate at the intermediary key value associated with both mapper and reducer. Such as:

$$(\text{tuple } \langle t, docid \rangle, \text{tf } f)$$

The above-mentioned term simplifies the key concept that it is just the only single record of the document while the containing value is foremost important as it indulge the term frequency (recurrence). This modification in the pseudo code guarantees that the key value either intermediary or single value they must anticipate in the correct order in the postings list.

It is necessary for the efficient information retrieval using inverted index that the similar term records precede to the similar reducer. The field term changes required that the algorithm should be modified in the same order as it occurs prior. The revised algorithm is described as follows.

By applying the value-to-key conversion design pattern, the execution framework is exploited to sort postings so that they arrive sorted by document id in the reducer.

```
1: class MAPPER
2:     method MAP(docid n, doc d)
3:         H ← new ASSOCIATIVEARRAY
4:         for all term t ∈ doc d do
5:             H{t} ← H{t} + 1
6:         for all term t ∈ H do
7:             EMIT(tuple ⟨t, n⟩, tf H{t})

1: class REDUCER
2:     method INITIALIZE
3:         t_prev ← ∅
4:         P ← new POSTINGSLIST
5:     method REDUCE(tuple ⟨t, n⟩, tf [f])
6:         if t ≠ t_prev ∧ t_prev ≠ ∅ then
7:             EMIT(term t_prev, postings P)
8:                 P.RESET()
9:             P.ADD(⟨n, f⟩)
10:            t_prev ← t
11:        method CLOSE
12:            EMIT(term t, postings P)
```

It could be noticed from above mentioned algorithm that the mapper became consistent in all the steps except of the some vital intermediary key values. The reducer now have been limited to the only one key in the data and single key holds single value in the revised algorithm mentioned before. The reducer is responsible for the sort order of the document id as new postings directly related to the posting list while using specified document id of the inverted index. . The map reduce role in the inverted index building is that as any new term is added to the document is directly keep in record the length as a side data to Hadoop file system. The advantage of the mapper considered as it in memory array used in document as the document length and score is added to the data set. The mapper array length needs to be initialized as an m length where m could be defined as a total number of mapper to make files representation more transparent to the system.

5.8 Comparison with naïve indexer

The inverted indexing in the previous algorithm is strengthening with the use of the intermediary key term value and keeping document is separate with the posting list

actual location. To examine the performance of the current algorithm for information retrieval system efficiency it is compared with naïve indexer as to check out the performance keeping in mind the size of the document, keywords and the time required to build the index for document. The selection of the naïve indexer is choosing as it shares the same input paradigm and keywords and documents structure. The naïve algorithm uses the state machine to tokenize the document as some query is parsing for the response and then match to the located document.

```
for all keyword do
  Build a vector v
  for all document do
    Add the row (keyword.id, document.id) into v
  end for
  Save v into the database
end for
```

The algorithm pseudo code demonstrate that the compared algorithm is more complex as it is viewed as an key word size n and document number size m and limitation of the size as an c. the naïve indexer is distinguish, it performs all database query and document action using the single transaction while the proposed algorithm performs all operation not to a single keyword but the whole document in the information retrieval system. The term t is denoted as a database latency which is greatly effect by the naïve indexer at the inverted indexing level. The time complexity is the common parameter to check out the performance of the algorithm. Following equation describe naïve indexer theoretical time complexity.

$$Time \propto n * (m * \bar{c} + \bar{t}) \qquad (1)$$

Following is the time complexity equation described under the proposed algorithm.

$$Time \propto m * (\bar{c} + \bar{t}) \qquad (2)$$

We can explains these equations as an document size c, keywords number n what more total number of documents m.

5.9 Document size

To test the proposed algorithm the keywords number are limited to the 1 thousand and document number 5 thousand to check out the proposed algorithm performance either it comprises to the varying length or remain slow while size and document size constantly changed. The above mentioned equation is supposed to be behaving in linear trends as the size grows performance remains consistent. Document indexing somehow do not comprise the small document very efficiently the proposed naïve indexer manages the small documents very efficiently for the sake of information retrieval system. The naïve indexer is also favorable as it scales the database in an efficient way as explained in the following examples of results and discussions.

5.10 Number of keywords

The proposed algorithm is tested with the given set of 1 thousand keywords and 5 thousand posting lists. The document size keywords are randomly increased at the size of indexing to check out the consistency in the performance of algorithm for information retrieval system. The equation number 1 explains that the naïve indexer will show minimum dependency on the document keywords.

6. Results and discussion

6.1 Construction of inverted index size

The purposed algorithm and inverted index contains the unique id and related posting in documents also sorted. There is a great deal of available documents techniques which may or may not be required the compression before the posting lists order presented to the reducer

6.2 Keywords

The keywords are greatly affected by the dictionary size so check out the performance of the inverted index builder used 10 thousand words of the language and different size of posting lists is used to check out the time required to build and design index for the document and then calculate total keywords of the dictionary. The following figure shows that the algorithm is linear with keywords of dictionary and is directly correlated with the average of the language dictionary keywords.

6.3 Corpus size

The corpus size along with dictionary keywords of 5 thousand words is used to test the performance of the system designed algorithm as the index and dictionary size increased it shows linear behavior.

6.4 Document size

The document size has also very strong impact on the performance of the indexing techniques. To evaluate the performance 5 thousand keywords and 5 thousand blog posts is used to check the behavior of the algorithm as the size of posts varies in the dynamically algorithm also proposed the linear trend with respect of the time complexity and average document as shown by the performance curve below. It is also explained that the size do not always shows the linear trend as the document size increase for example for the single instance it create more than one entries in the document index To test the proposed algorithm the keywords number are limited to the 1 thousand and document number 5 thousand to check out the proposed algorithm performance either it comprises to the varying length or remain slow while size and document size constantly changed. The above mentioned equation is supposed to be behaving in linear trends as the size grows performance remains consistent. Document indexing somehow do not comprise the small document very efficiently the proposed naïve indexer manages the small documents very efficiently for the sake of information retrieval system. The naïve indexer is also favorable as it scales the database in an efficient way as explained in the following examples of results and discussions.as explained below.

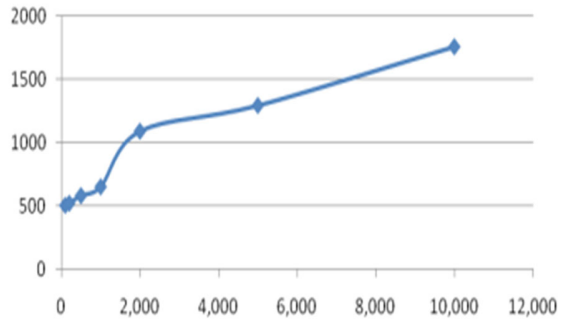

Figure 8: time explained with the respect of the dictionary keywords. Time is at y axis while keywords are at the x axis.

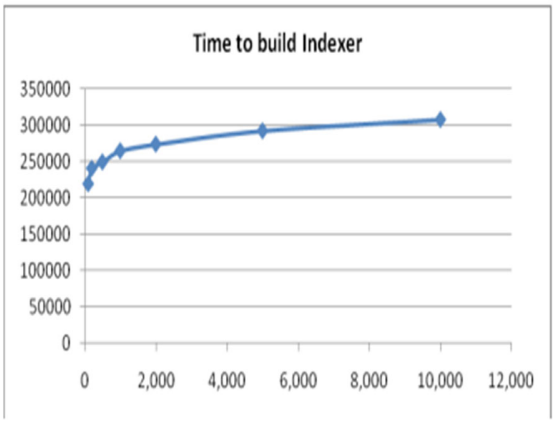

Figure 9: time explained with the respect of the dictionary keywords. Time is at y axis while keywords are at the x axis.

The above explained the time required to build corpus keywords and also explained that the algorithm is totally exceptionally independent from the number of keywords and the dictionary size resides or located in the document files of the information retrieval system. Time complexity became exceptionally very low as the size of the keywords increased. With the explanation of fig 1 and 2 it is very obvious to keep in record that fig 2 increases the time complexity to indicate that the as the number of keywords increase database index construction also needs to be update accordingly. It is the property of the proposed algorithm that the time magnitude is in 2 order and exceptionally take less time than index building. It validates the use of the algorithm for information retrieval efficiency.

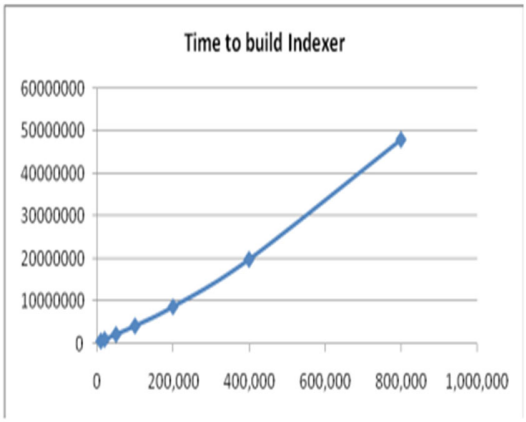

Figure 10: Explanation of building inverted index as the document needed to be indexed. The Document size is explained at the x axis and the time in ms (millisecond) at the y axis of the graph.

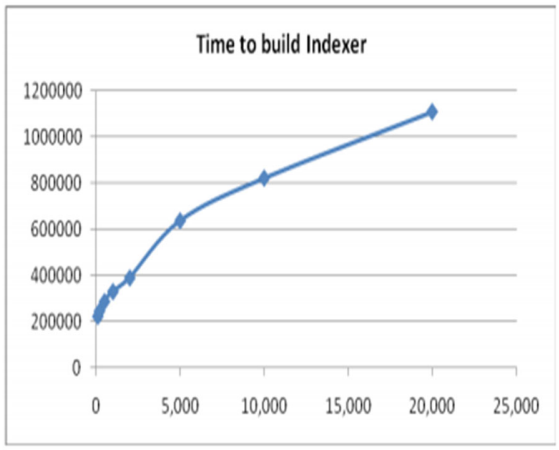

Figure 11: Explanation of building inverted index for the average document needed to be indexed. The Document size is explained at the x axis and the time in ms (millisecond) at the y axis of the graph.

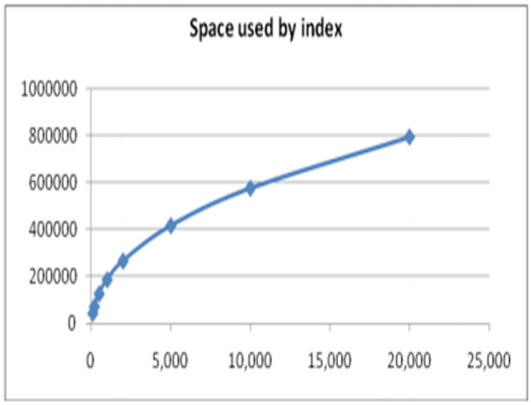

Figure 12: Explanation of building inverted index as the document needed to be indexed. The Document size is explained at the x axis and the time in ms (millisecond) at the y axis of the graph. The document rows record is indexed for the explanation of the graph.

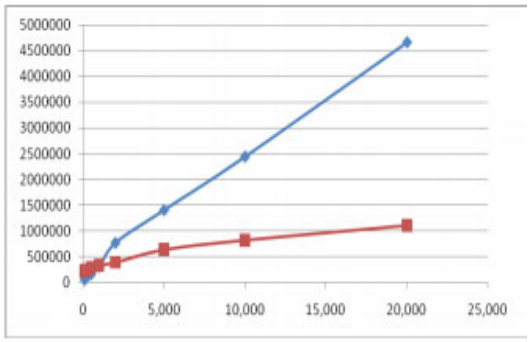

Figure 13: Explanation of building inverted index for the average document needed to be indexed. The Document size is explained at the x axis and the time in ms (millisecond) at the y axis of the graph.

7. Conclusion and Future work

Excessive use of technology has increased data volume at a very high rate due to information retrieval system need to very efficient. For this scalable and efficient algorithms for documents and indexed construction is defined within the available resources of hardware keeping in mind the availability of that resources to the common man block sort indices and standalone pass in memory are algorithm that are supportable for a common man. This algorithm map reduce the extra steps to reduce required to search on external hard drive and without wasting time reviewing non relevant document. In future Object model data bases used for specific indexing based solution on data which is queried.

8. Acknowledgment

Authors are grateful to Dr. Muhammad Irfan Khan, Department of Computer Science, GC University Faisalabad for his valuable suggestions in the analysis of data.

8. References

[1] Bai, Qiuying & Ma, Chi and Chen, Xuechang. *A new index model based on inverted index. ICSESS 2012* - Proceedings of 2012 IEEE 3rd International Conference on Software Engineering and Service Science. 157-160. 2012.

[2] Chandwani, G., Ahlawat, A., & Dubey, G. *An approach for document retrieval using cluster-based inverted indexing*. Journal of Information Science. 2021

[3] Junxiu, A. *The Research of Non-back Multi-words Matching Algorithm Based on Aggregate Address Inverted Index*. In 2009 International

Conference on E-Learning, E-Business, Enterprise Information Systems, and E-Government (pp. 200-203). IEEE.2009.

[4] Al-Dallal, A., and Shaker, R. *Genetic algorithm in web search using inverted index representation.* In 2009 5th IEEE GCC Conference & Exhibition (pp. 1-5). IEEE. 2009.

[5] Siregar, A. M., and Puspabhuana, A. *Improvement of term weight result in the information retrieval systems.* In 2017 4th International Conference on New Media Studies (CONMEDIA) (pp. 108-112). IEEE.2017.

[6] Orhean, A. I., Ijagbone, I., Raicu, I., Chard, K., and Zhao, D. *Toward scalable indexing and search on distributed and unstructured data.* In 2017 IEEE International Congress on Big Data (BigData Congress) (pp. 31-38). IEEE. 2017.

[7] Cambazoglu, B. B., Kayaaslan, E., Jonassen, S., and Aykanat, C. *A term-based inverted index partitioning model for efficient distributed query processing.* ACM Transactions on the Web (TWEB), 7(3), 1-23.2013

[8] Jo, T. *Clustering news groups using inverted index based NTSO*. In 2009 First International Conference on Networked Digital Technologies (pp. 1-7). IEEE. 2009.

[9] Teshome, A. K., Kibret, B., and Lai, D. T. *A review of implant communication technology in WBAN: Progress and challenges*. IEEE reviews in biomedical engineering, 12, 88-99.2018.

[10] Cheng, X., and Singer, A. *The spectrum of random inner-product kernel matrices.* Random Matrices: Theory and Applications, 2(04).2013.

[11] Anh, V.N., Moffat, A. *Inverted Index Compression Using Word-Aligned Binary Codes.* Information Retrieval 8, 151–166. 2005.

[12] Zobel, J., and Moffat, A. *Inverted files for text search engines.* ACM computing surveys (CSUR), 38(2), 2006.

[13] Jung, W., Roh, H., Shin, M., and Park, S. *Inverted index maintenance strategy for flashSSDs: Revitalization of in-place index update strategy.* Information Systems, 49, 25-39.2015.

[14] Dabbèchi, H., Haddar, N., Abdallah, M. B., and Haddar, K. *A unified multidimensional data model from social networks for unstructured data analysis.* In 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA) (pp. 415-422). IEEE.2017.

[15] Jayaraman, P. P., Mitra, K., Saguna, S., Shah, T., Georgakopoulos, D., and Ranjan, R. *Orchestrating quality of service in the cloud of things ecosystem.* In 2015 IEEE International Symposium on Nanoelectronic and Information Systems. pp. 185-190. 2015.

[16] Sun, S., Gong, J., Zomaya, A. Y., and Wu, A. *A distributed incremental information acquisition model for large-scale text data*. Cluster computing, 22(1), 2383-2394.2019.

[17] Nepomnyachiy, S., and Suel, T. *Efficient index updates for mixed update and query loads*. In 2016 IEEE International Conference on Big Data (Big Data) (pp. 984-991). IEEE.2016.

[18] Giangreco, I., Al Kabary, I., and Schuldt, H. *Adam-a database and information retrieval system for big multimedia collections.* In 2014 IEEE International Congress on Big Data (pp. 406-413). IEEE.2014.

[19] Lomotey, R. K., and Deters, R. *Towards knowledge discovery in big data*. In 2014 IEEE 8th International Symposium on Service Oriented System Engineering (pp. 181-191). IEEE. 2014.

[20] Arab, A., and Abrishami, S. *MDMP: a new algorithm to create inverted index files in BigData, using MapReduce.* In 2017 7th International Conference on Computer and Knowledge Engineering (ICCKE) (pp. 372-378). IEEE. 2017.

[21] Ma, C., Xia, W., Chen, F., Liu, J., Dai, Q., Jiang, L., and Liu, W. *A content-based remote sensing image change information retrieval model*. ISPRS International Journal of Geo-Information, 6(10), 310. 2017.

[22] Lomotey, R. K., and Deters, R. *Architectural designs from mobile cloud computing to ubiquitous cloud computing-survey.* In 2014 IEEE World Congress on Services (pp. 418-425). IEEE.2014.

[23] Lomotey, R. K., and Deters, R. *Analytics-as-a-service framework for terms association mining in unstructured data*. International Journal of Business Process Integration and Management, 7(1), 49-61.2014.

[24] Konow, R., Navarro, G., Clarke, C. L., and López-Ortíz, A. *Faster and smaller inverted indices with treaps.* In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval (pp. 193-202).2013.

[25] Jiang, D., Leung, K. W. T., Yang, L., & Ng, W. TEII: *Topic enhanced inverted index for top-k document retrieval.* Knowledge-Based Systems, 89, 346-358.2015.

[26] Sung W, Ahamd J., Muhammad, K., Bakshi, S., and Baik. *Object-oriented convolutional features for fine-grained image retrieval in large surveillance datasets*. Future Generation Computer Systems, 81, 314-330. 2018.

[27] Giridharan, J., and Vairavan, S. V. *Inverted index and interval lists for keyword search.* In 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE) (pp. 1-4). IEEE. 2014

[28] Wu, H., Li, G., and Zhou, L. Ginix: *Generalized inverted index for keyword search*. Tsinghua Science and Technology, 18(1), 77-87.2013

[29] B. Wang, W. Song, W. Lou and Y. T. *Hou, Inverted index based multi-keyword public-key searchable encryption with strong privacy guarantee,* IEEE Conference on Computer Communications (INFOCOM), 2015, pp. 2092-2100.

[30] A. Babenko and V. Lempitsky, *The Inverted Multi-Index,* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 6, pp. 1247-1260, 2015.

[31] Y. Qiao, X. Yun and Y. Zhang, *Fast Reused Function Retrieval Method Based on Simhash and Inverted Index,* 2016 IEEE Trustcom/BigDataSE/ISPA, 2016, pp. 937-944, 2016.

[32] Lin, Z., Ding, G., Han, J., and Wang, J. *Cross-view retrieval via probability-based semantics-preserving hashing.* IEEE transactions on cybernetics, 47(12), 4342-4355. 2016.

[33] Dell, L. A., Patzke, N., Spocter, M. A., Siegel, J. M., and Manger, P. R. *Organization of the sleep-related neural systems in the brain of the harbour porpoise (Phocoena phocoena).* Journal of Comparative Neurology, 524(10), 1999-2017. 2016

[34] Orhean, A. I., Pop, F., and Raicu, I. *New scheduling approach using reinforcement learning for heterogeneous distributed systems.* Journal of Parallel and Distributed Computing, 117, 292-302. 2018

[35] S. Anggai, I. S. Blekanov and S. L. Sergeev, *Construction inverted index for dynamic collections visualization in thematic virtual museums system*, 2017 3rd International Conference on Science and Technology - Computer (ICST), 2017, pp. 186-189.

[36] Gamberini, L., Barresi, G., Maier, A., and Scarpetta, F. *A game a day keeps the doctor away: A short review of computer games in mental healthcare.* Journal of CyberTherapy and Rehabilitation, 1(2), 127-145. 2008.

[37] Zhu, N., Lu, Y., He, W., & Hua, Y. *A content-based indexing scheme for large-scale unstructured data.* In 2017 IEEE Third International Conference on Multimedia Big Data (BigMM) (pp. 205-212). IEEE. 2017.

[38] Singh, R., and Mohaar, G. S. *Fast Document Indexing Using Aho-Corasick State Machine.* In 2016 IEEE 17th International Conference on Information Reuse and Integration (IRI) (pp. 469-475). IEEE.2016.

[39] Karthika, N., and Janet, B. *Word pair index structure for information retrieval using Terrier3. 5.* In 2017 International Conference on Computational Intelligence in Data Science (ICCIDS) (pp. 1-6). IEEE. 2017.

[40] Jing, W., Tong, D., Chen, G., Zhao, C., and Zhu, L. *An optimized method of HDFS for massive small files storage.* Computer Science and Information Systems, 15(3), 533-548. 2018

[41] Do, Y., Kim, S. H., Na, I. S., Kang, D. W., and Kim, J. H. *Image retrieval using wavelet transform and shape decomposition.* In Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication (pp. 1-8).2013.

[42] Wei, Z., and Jinzhe, J. *An improved association rule algorithm based on trie and inverted index.* In Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE) pp 1669-1672. IEEE.2011.

[43] Kori, S., Zhu, Y., Yamaguchi, K., Takiguchi, S., and Takama, Y. *Analysis of user's behaviour based on search intentions for information retrieval using search engines.* In 2015 Conference on Technologies and Applications of Artificial Intelligence (TAAI), (pp. 64-70). IEEE. 2015

[44] Abdullah, A., Yong, K. K., Karuppiah, E. K., and Chong, P. K. *Multi keyword range search in gpu and mic: A comparison study*. In 2014 IEEE Conference on Open Systems (ICOS). (pp. 117-122). IEEE.2014

[45] Liu, M., Po, L. M., Rehman, Y. A. U., Xu, X., Li, Y., and Feng, L. *A novel inverted index file based searching strategy for video copy detection.* In 2017 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA) (pp. 307-312). IEEE. 2017

[46] Li, G., Ooi, B. C., Feng, J., Wang, J., and Zhou, L. *Ease: an effective 3-in-1 keyword search method for unstructured, semi-structured and structured data.* In Proceedings of the 2008 ACM SIGMOD international conference on Management of data (pp. 903-914).2008

[47] Li, X., Li, K., Qiao, D., Ding, Y., and Wei, D. *Application research of machine learning method based on distributed cluster in information retrieval.* In 2019 International Conference on Communications, Information System and Computer Engineering (CISCE) (pp. 411-414). IEEE.2019.

[48] Zamani, H., Dehghani, M., Croft, W. B., Learned-Miller, E., and Kamps, J. *From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing.* In Proceedings of the 27th ACM international conference on information and knowledge management (pp. 497-506). 2018.

[49] Park, B. K., and Song, I. Y.*Toward total business intelligence incorporating structured and unstructured data.* In Proceedings of the 2nd International Workshop on Business intelligence and the WEB (pp. 12-19).2011.

[50] Hariharan, R., Hore, B., Li, C., and Mehrotra, S. *Processing spatial-keyword (SK) queries in geographic information retrieval (GIR) systems.* In 19th International Conference on Scientific and Statistical Database Management (SSDBM 2007) (pp. 16-16). IEEE.2007