International Journal of Computer Engineering and Applications, Volume IX, Issue VI, Part I, June 2015
www.ijcea.com ISSN 2321-3469

SEARCH ENGINE OPTIMIZATION USING DATA MINING APPROACH

Khattab O. Khorsheed [1], Magda M. Madbouly [2], Shawkat K. Guirguis [3]

[1,2,3] Department of Information Technology, Institute of Graduate Studies and Researches, Alexandria University, 278 Geish Street, Stanley, Alexandria, Egypt

ABSTRACT:

Search Engine Optimization (SEO) is the procedure used to improve the visibility of the results searched for on a free search engine for a website or a web page. The optimization targets different types of items such as images, videos, academic articles, etc. The SEO can also be defined as the process affecting the visibility of a website or a webpage in search engines. This paper aims to improve the search time of search engines to the greatest extent using the K-Means Algorithm which does the Clustering of the database. Upon conducting experiments, it is found that the algorithm produces better results without the clustering. As the size of data increases, the time used in searching is affected as search time increases to a great extent and the search process becomes slower. The paper decided to use the Message Passing Application Programming Interface(MPAPI) technique with the K-Means to solve the delay problem during search. Using this technique, search takes place in more one cluster in parallel. Each thread can connect to its own case; connection takes place through messages not State transfer between Threads. We then compare the results of the normal search results (Sphider) and the results that come out of using K-Means and MPAPI.

Keywords: Search Engine Optimizations (SEO), Data Mining, Clustering, K-Means Algorithm, Message Passing Application Programming Interface (MPAPI), Sphider, Vector Space Model (VSM)

[1] INTRODUCTION

Web-based search engines (e.g. Google, Bing, Baidu) index and rank information on the web such as documents, images, videos and so on. The search results are listed based on a certain ranking algorithm starting with the most related ones up front. Web-based search engines are rated according to their effectiveness in retrieving the closely relevant information over the internet with efficiency; for instance, Google has done a great job in improving both the effectiveness of information retrieval and the efficiency of the query performance. However, with the increasing popularity of social networks, most of the recent research about general web-based search engines have relied on integrating the social connectivity factors into the search results, hence the name 'social search,' for instance, a semantic social search engine that utilizes social networks like Google+, Twitter and Facebook [1].

Search Engine Optimization (SEO) is a method to get a better ranking for a website in search engines such as Google, Yahoo or Bing. A search engine optimization campaign pairs on-site optimization with off-site tactics which means that one makes changes to the site itself while they build a portfolio of natural looking back links to increase their organic rankings. When internet users search for the product or service related, the website relevance to specific keywords which internet users search for online [2]. The process of optimizing search engine includes researching keywords, creating content, building links and making sure that the website is visible in the search engines. According to incomplete statistics, the number of Internet users in China in 2008 reached 200 million people, 89% of those internet users mainly use the internet to obtain information, of which 88.8% will make use of internet search engines search information, second only to send and receive e-mail. Web search engines have become an important part. Searching some kind of information on the web became hectic [3].

The complete SEO procedure works with two types of optimization techniques, on-page and off-page SEO optimization. Both techniques have their personal, discrete and extensive processes to rank websites on top of search engines. The SEO process starts with on-page SEO optimization. Just the once the whole on-page SEO optimization is complete, the off-page SEO optimization starts. Off-page SEO includes tricks which are chosen to make relevant back links towards the website to make the web page appropriately in front of search engine spiders. Second off-page SEO is doing the responsibility to improve our site's search engine rankings outside of our site. The only thing one can do off-site to increase the rankings is building up more links [4].

SEO Benefits: Popularity of this technique popularity will increase. Increase Visibility once a website has been optimized, it will increase the visibility of a website in search engine. More people will visit the website. Targeted traffic Search Engine Optimization can increase the number of visitors to the website for the targeted keywords. High ROI (Return of Investments) an effective SEO campaign can bring a higher return on investment than any other marketing. It will increase the volume of sales. Online Marketing and Promotion: best strategy for promotion [5].

There are many challenges that have to be considered when working in the field of SEO as the search engine could give out too many web-pages in the output, the difficulty in observation of both the output and outcome may result in a conflict between focusing on the observable parts of the work that can be readily measured and reported to meet the accountability requirements versus the work with less tangible output and outcome; thus, the users would have to spend more time finding their desired information from the long search result list. All existing search engines adopt several techniques and approaches to improve the performance of the search engines. However, after evaluating the performance of search engines based on the retrieved web contents, it is apparent that only a few attempts were made to restructure the query, providing alternate queries or personalizing the web search [6].

Data mining represents the integration of several fields, including machine learning, database systems, data visualization, statistics and information theory. Data mining can be defined as a non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. It employs techniques from machine learning, statistics, and databases [7].

Clustering is an unsupervised algorithm, which requires a parameter that specifies the number of clusters k. For setting this parameter either requires detailed knowledge of the data set or requires the algorithm to be run for different values of k to determine the correct number of clusters. However, for large and multidimensional data process of clustering becomes time consuming and determining the correct number of clusters in large data becomes difficult [8].

K-Means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problems. The procedure follows a simple and easy way to classify the given data set through a certain number of clusters (assume k clusters) [9]. The main idea is to define k centroids, one for each cluster K-Means algorithm features quick clustering and easy operation, and is applied to the cluster analysis of several data such as texts, images and others; however, this algorithm tends to terminate iterative process quickly to only obtain a partial optimal results, and fluctuate the clustering result because of random selection of the initial iterative center point. Due to the fact that the clustering is often applied to the data of the cluster quality the end-users can’t judge and this fluctuation is difficult to be accepted in the application, it is of great significance to improve the quality and stability of clustering results in the analysis of the text cluster [10].

The Message Passing model (MPAPI) is intended as a standard implementation of the "message passing" model of parallel computing. A parallel computation consists of a number of processes, each working with some local data. Each process has purely local variables, and there is no mechanism for any process to directly access the memory of another. Sharing of data between processes takes place by message passing, that is, by explicitly sending and receiving data between processes. Note that the model involves processes, which need not, in principle, be running on different processors. In this course, it is generally assumed that different processes are running on different processors and the terms "processes" and "processors" are used interchangeably (e.g., by speaking of processors communicating with one another) [11].

In this paper, we have improved the search engine Sphider which is a popular open-source web-spider and search engine. Sphider includes an automated crawler which can follow the links found on a website, and an indexer which builds an index of all the search terms found in the pages and uses the Porter Stemming Algorithm in a process of removing the commoner morphological and database by using K-Means Algorithm and MPAPI to reduce the time of clustering all the keywords.

The rest of the paper is organized as such:

- Section 2: Related Work for Search Engine Optimization
- Section 3: Description of the Proposed System
- Section 4: Test Results and Discussion of the Meaning
- Section 5: Short Summary and Outlook of Future Works

[2] RELATED WORK

This section presents the research progress and findings on techniques and algorithms for search engine optimization. K. K. Kattamuri et al. They focused on two different techniques

combined for fast search retrieval [12]. In this system, we have used various techniques like Vector Space Indexing, Stemming, Cosine Measure Ranking and K-Means clustering for efficient search and retrieval. With Vector Space Indexing, all the documents in the database will be indexed against terms in each document. The problem arises here as it takes a great amount of time in order to process all the terms step-by-step. In order to reduce terms for indexing, we have used the Stemmer technique that optimizes the terms and created an optimized way of searching documents by Clustering which reduces the searching time by limiting the irrelevant documents. For this, using K-means clustering algorithm was a better algorithm for differentiating clusters.

In this method, the researcher used an initial value of 'k' as a means of clustering. However, finding the value of 'k' accurately, will not be possible and that will affect the entire clustering process. They propose a new technique where the value of 'k' is updated in an incremental procedure, depending on the number of documents in the cluster. With these techniques, Search Engine will be capable of producing accurate results.

For a search engine, a step-by-step procedure of indexing the documents, performing search and producing the results will take effort and time. Thus, the need of a technique which can perform all the steps at a time is growing. The concept of Parallel Processing perfectly works here as it processes all the works simultaneously. Peng Jiang, et al. proposed K-means approach based on the concept hierarchical tree to cluster search results [13]. This algorithm not only overcomes weaknesses of the classic K-means method: the results produced depend on the initial seeds and the parameter 'k' is often unknown, but also satisfies the requirements of online search results clustering. they method utilizes the semantic relationships between documents by mapping terms of concepts, in the concept hierarchical tree, which can be constructed by word net. Have used the developed meta-search and clustering system based on our approach, followed by using an impersonal and repeatable evaluation solution. The experimental results indicate that our proposed algorithm is effective and suitable in performing the task of clustering search results. A CHT K-Means algorithm which is suitable to cluster web search results since it incorporates the implicit concepts in web documents and utilizes the semantic relation between them. A meta-search and clustering system based on our algorithm has been developed to verify its validity. In addition, the evaluation solution used is impersonal and repeatable. Experimental results show that our algorithm is effective and suitable for practical applications of online search results clustering. Cui and Hu highlight the specific requirements for optimizing the search queries, and present a novel website building and design concepts based on the empirical research pertaining to internal coding methods and website contents. In addition, the authors elaborate search engine optimization tools and strategies specific to the e-commerce sites for the sake of effective website promotion [14].

When a user searches a website through an optimized search engine, then the entire website can attain a higher ranking position. This improves the website traffic and enhances the website sales capability which brings employing specific tools, strategies and search-engine friendly methods as a priority.

The four techniques used for the Tools of SEO category are:

- Keyword tools

- Link Tools
- Usability Tools
- High-quality Incoming Links

For the Strategies for SEO, three methods are elaborated, which are:

- Website structure
- Space strategy
- Writing website titles strategy

The methods used in the Friendly Methods of SEO category are structured optimization of frames, optimization of images, URLs, directory structures, navigation of website, optimization of flash and web form optimization.

[3] PROPOSED MPAPI WITH K-MEANS CLUSTERING

Our work to improve the search engine Sphider is a popular open-source web spider and search engine. It includes an automated crawler, which can follow the links found on a site, and an indexer which builds an index of all the search terms found in the pages and use Porter Stemming Algorithm is a process for removing the commoner morphological and in flexional endings from words in English. We took all the words in database our search engine (sphider) using text Documents Clustering using K-Means Algorithm, and Implementation Message Passing Application Programming Interface (MPAPI) to reduce the time of clusters all keywords and reduce time of search. Sphider is a popular open source web spider and search engine which includes an automated crawler which can follow the links found on a specific website and an indexer that sets all the search terms found on the pages in an index. The Porter Stemming Algorithm is a process made to remove the commoner morphological and in flexional endings from words in English. We applied Clustering using K-means Algorithm to text documents and MPAPI to take all the words in the database of the search engine in order to reduce the time of clustering keywords and search time.

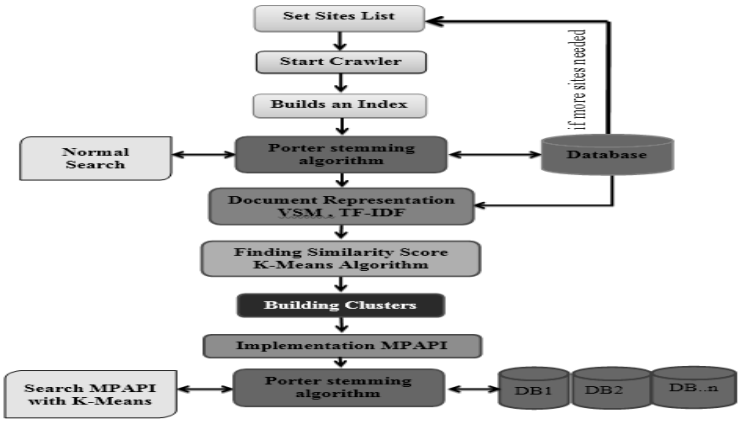

Figure: 1. Proposed MPAPI with K-Means Clustering

[3.1] SEARCH ENGINE

Search engines are programs used to find documents or specific information on the World Wide Web. In this paper, we develop the Sphider search engine which is a lightweight web spider and search engine written in PHP, using MySQL as its backend database. It's widely used to add search functionality to websites or building customized search engines as it is easy to set up and modify thousands of websites worldwide use it. Sphider supports all standard search options, but also includes a plethora of advanced features such as word auto completion, spelling suggestions and so on. The sophisticated administration interface makes administering the system easy.

Sphider uses word Porter stemming algorithm stemming, a process which reduces inflected or derived words from their stem, base or root form. It forms an initial step in most NLP-IR techniques, especially search and indexing algorithms. Natural language texts typically contain many different variations of a basic word; morphological variants (e.g., COMPUTATIONAL, COMPUTER, COMPUTERS, COMPUTING etc.) are generally the most common, with other sources including valid alternative spellings, misspellings and variants arising from transliteration and abbreviation. This is supposed to increase the effect of searching. Porter stemming is used in Vector Space Search Engines to reduce the term-space while maintaining the semantic content of term space. Vector space model (or term vector model) is an algebraic model for representing text documents (and any objects, in general) as vectors of identifiers, such as, for example: index terms. A document is represented as a vector and each dimension corresponds to a separate terminal. It is used in information filtering, information retrieval, indexing and relevancy rankings. The keyword-based similarity function is defined as the following:

$$Similarity_{keyword}(p,q) = KN(p,q)/MAX(kn(p), kn(q)) \quad (1)$$

Where $kn(p)$ is the number of keywords in a query, $KN(p,q)$ is the number of common keywords in two queries. If query terms are weighted, the following modified formula can be used instead:

$$Similarity_{w-keyword}(P,q)=\sum_{i=1}^{N}(w\big(k_i(p)\big) + w(k_i(q)))/Max(kn(p), kn(q) \quad (2)$$

Where $w\big(k_i(p)\big)$ is the weight of the i-th common keyword in query p and $kn(q)$ becomes the sum of weights of the keywords in a query. In our case, we use $tf * idf$ for keyword weighting.

[3.1.1] CRAWLER AND INDEXER

Web search engines and other sites use web crawling or spider software to update their web content of others sites' web content. Web crawlers can copy all the pages they visit for later processing by a search engine that indexes the downloaded pages so that users can search them faster. A web crawler starts with a list of URLs to visit, called the seeds. As the crawler visits these URLs, it identifies all the hyperlinks in the page and adds them to the list of URLs to visit;

that list is called the crawl frontier. URLs from the frontier are recursively visited according to a set of policies. If the crawler is performing archiving of websites, it copies and saves the information as it goes. Those archives are usually stored in a way so they can be viewed, read and navigated as if they were on the live web, but are preserved as 'snapshots'.

The indexer processes the pages crawled by the crawler. First, it chooses which pages to index, for instance, it might discard duplicate documents, and then it creates different auxiliary data structures. Most search engines build some variant of an inverted index data structure for the word 'text index' and links 'structure index'. The inverted index contains for each word a sorted list of couples, such as doc ID and position in the document. It's particularly designed and optimized for indexing files. Using the index built by the indexer, the search engine can access almost directly to sections of the database which contains the information a user is looking for. Search engine ranking depends a lot upon the website indexing. The more of the website's web pages include (indexed) by search engine than it will have a better search engine ranking. Search engine practitioner's one of primary purpose is the website indexing so that every desired web page should be indexed.

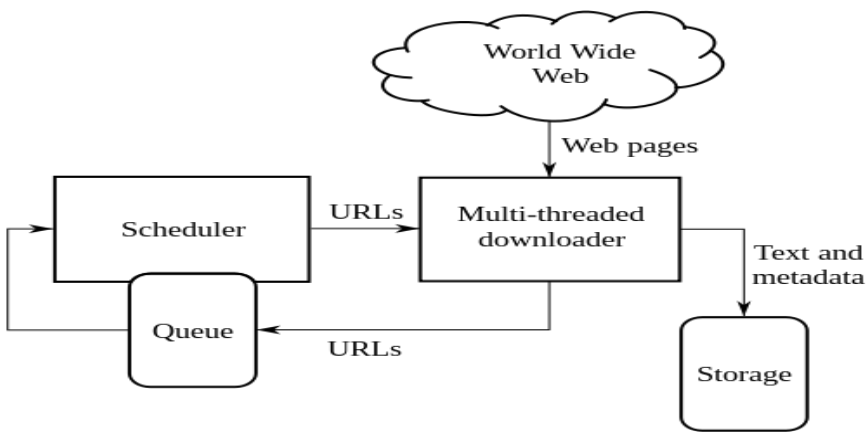

Figure:2. High-level Architecture of a Standard Web Crawler

[3.2] TEXT DOCUMENTS CLUSTERING USING K-MEANS ALGORITHM

[3.2.1] DOCUMENT REPRESENTATION

Each document is presented as a vector using the vector space model. The vector space model, also called term vector model is an algebraic model for presenting text documents (or any object, in general) as vectors of identifiers. Documents and queries are presented as vectors.

$$d_j = (w_{1,j}, w_{2,j}, \ldots\ldots, w_{t,j}), q = (w_{1,q}, w_{2,q}, \ldots\ldots, w_{n,q})$$

Each dimension corresponds to a separate terminal. If a term occurs in the document, its value in the vector is non-zero. There are several different ways of computing these values known as 'term' weights which have been developed. One of the best known schemes is TF-IDF weighting. The definition of the term depends on the application. Typical terms are single

words, keywords, or long phrases. If words are chosen to be the terms, the dimensionality of the vector is the number of words in the vocabulary (the number of distinct words occurring in the corpus). Vector operations can be used to compare documents with queries. Here we have defined the document vector class which holds the document and its corresponding representation of vector space, and the instance of a document collection represents all the documents to be clustered.

[3.2.2] TERM FREQUENCY–INVERSE DOCUMENT FREQUENCY (TF-IDF)

Term Frequency-Inverse Document Frequency method or TF-IDF, is a method which presents important words or terms in documents. VSM method is a method to measure similarity value of a document to another document based on the weight values of terms which are taken from the TF-IDF method. TF-IDF and VSM methods are usually used for sorting and counting the similarities in documents. This ability can also be used to build a classification system of negative or not negative websites.

The TF-IDF method will count the weight of WI in a document d from term frequency value or $TF\ (t,d)$ and document frequency value or $DF\ (t).TF\ (t,d)$ is the appearance frequency of word t in the document d, which is mathematically formulated by Eq. 3:

$$TF\ (t,d)\ =\ \sum (tk,dj) \tag{3}$$

$DF\ (t)$ is amount of documents which contains word t. Thus, Inverse Document Frequency (IDF) can be counted from a number of documents $|D|$ divided by number of documents which contain word it or DF (t) and mathematically is formulated by Eq. 4:

$$IDF\ (T)\ =\ Log\left(\frac{|D|}{DF(t)}\right) \tag{4}$$

In TF-IDF method, weighs Wi is the $TF(t,d)$ values multiplied by $IDF(t)$ values. The values of weights Wi can be counted using Eq. 5:

$$Wi\ =\ TF\ (t,d)\ \times\ IDF(t) \tag{5}$$

According to Equations 1 and 2, the values of weights Wi can be found using Eq. 6:

$$Wi\ =\ TF\ (t,d)\ X\ Log\left(\frac{|D|}{DF(t)}\right) \tag{6}$$

Vector Space Model or VSM is a method which counts number of similarities of document to another document. In VSM algorithm, document and query were assumed as vectors that occur in n–dimensional. VSM in text classification will count the similarity value between a document not known what category it is with other documents known what category it is using

the training output. VSM finds the value of cosine angle between two vectors based on values of weights we got from TF-IDF calculation. Cosine or similarity values can be found using Eq. 7:

$$sim(d_i\,, c_j) = \frac{\sum_{k=1}^{n}(w_{ik}\,.w_{jk})}{\sqrt{\sum_{k=1}^{n} w_{ik}^2 \cdot \sum_{k=1}^{n} w_{jk}^2}} \tag{7}$$

W_{ik} and W_{jk} are the values of the weights from feature extraction of the word or term k in a text DI and category cj. For example, given $W_{ik} = \{a_1, a_2, a_3, \dots, a_n\}$ and $W_{jk} = \{b_1, b_2, b_3, \dots, b_n\}$ then $sim(di, cj)$ is $(W_{ik}\,.W_{jk}\,) = \{a_1.b_1, a_2.b_2, a_3.b_3, \dots, a_n.b_n\}$ divided by the square root of $\{a_{\,1\,2}, a_{22}\,, a_{32}\,, \dots, a_{n\,2}\,\}$ multiplied by $\{b_{12}\,, b_{2\,2}\,,\ b_{3\,2}\,, \dots, b_{n\,2}\}$.

[3.2.3] FINDING SIMILARITY SCORE

The cosine similarity was used to compare the similarities of the document. The method takes two arguments (vec A) and (vec B) as parameter which are a vector representation of document A and B, and returns the similarity score which lies between 1 and 0, indicating that document A and B are completely similar and dissimilar respectively. Relevance rankings of documents in a keyword search can be calculated, using the assumptions of document similarities theory, by comparing the deviation of angles between each document vector and the original query vector where the query is represented as the same kind of vector as the documents. In practice, it is easier to calculate the cosine of the angle between the vectors, instead of the angle itself as shown in Eq. 8:

$$\cos\theta = \frac{d_2\,.q}{\|d_2\|\,\|q\|} \tag{8}$$

Where $\boldsymbol{d_2}\,.\boldsymbol{q}$ is the intersection (i.e. the dot product) of the document (d2 in the figure to the right) and the query (q in the figure) vectors, $\|\boldsymbol{d_2}\|$ is the norm of vector d2, and $\|\boldsymbol{q}\|$is the norm of vector q. The norm of a vector is calculated as shown in Eq. 9:

$$\|\boldsymbol{q}\| = \sqrt{\sum_{i=1}^{n} q_i^2} \tag{9}$$

[3.2.4] K-MEANS ALGORITHM IMPLEMENTATION

To implement K-Means algorithm we have defined a class Centroid in which documents are assigned during the clustering process. The k-means algorithm uses the vector space model by iteratively optimizing k centroid vectors, which represent clusters. These clusters are updated by taking the mean of the nearest neighbors of the centroid. The algorithm proceeds to iteratively optimize the sum of squared distances between the centroids and the nearest neighbor set of vectors (clusters). This is achieved by iteratively updating the centroids to the cluster means and reassigning nearest neighbors to form new clusters, until convergence. The centroids are initialized by selecting k vectors from the document collection uniformly at random. It is

well known that k-means is a special case of Expectation Maximization with hard cluster membership and isotropic Gaussian distributions

The K-Means algorithm aims to partition a set of objects, based on their attributes/features, into k clusters, where k is a predefined or user-defined constant. The main idea is to define k centroids, one for each cluster. The centroid of a cluster is formed in such a way that it is closely related (in terms of similarity function; similarity can be measured by using different methods such as cosine similarity, Euclidean distance, Extended to all objects in that cluster. The overall heuristically process is shown in the following list:

1. Randomly select k of the objects, each of which initially represents a cluster mean or Centroid.
2. For each of the remaining objects, an object is assigned to a cluster to which it is the most similar, based on the distance between the object and the cluster mean. It then computes the new mean for each cluster using Eq. 10:

$$M_j = \frac{1}{n_j} \sum_{\nabla z_p \in c_i} Z_P \qquad (10)$$

Where, Mj is the centroid of cluster j and Nj is the number of data points in cluster j.
3**.** This process iterates until the criterion function converges. Typically, the square – error criterion is used, defined using Eq. 11:

$$E = \sum_{i=1}^{k} \sum_{p \in c_i} |p - m_i| \qquad (11)$$

Where p is the data point and mi is the center for cluster Ci. E is the sum of squared error of all points in the dataset. The distance of criterion function is the Euclidean distance which is used to calculate the distance between data points and cluster center. The Euclidean distance between two vectors

$x = (x_1, x_2, x_3, \dots, xn)$ and $y = (y_1, y_2, y_3, \dots . y_n)$ can be calculated using Eq.12:

$$d(x_i, y_i) = \sum_{i=1}^{n} \sqrt{(x_i - y_i)^2} \qquad (12)$$

[3.3] MESSAGE PASSING APPLICATION PROGRAMMING INTERFACE (MPAPI)

Message passing is a different approach to concurrency than we are used to when using modern, imperative languages like C++, C# or Java. In those languages we use shared state concurrency, where all threads in the same process space has access to the same areas of memory (i.e. variables). A problem, thus, arises with how we synchronize the threads to avoid inconsistent

memory and is often done with locks, semaphores, monitors and other constructs. As explained in the preface this is not as trivial as one might think, and there are always one or more problems in the synchronization that can be very hard to detect, leading to unstable systems and a lot of time spent on debugging and rewriting parts of the software.

In message passing concurrency each thread has access only to its own state. This helps programmers write more robust software since the need to synchronize memory no longer exists. The only means for a thread to communicate with other threads is by sending them messages, and state is not transferred between threads in these messages. MPAPI works as multithreaded software. Multithreading is mainly found in multitasking operating systems and is a widespread programming and execution model that allows multiple threads to exist within the context of a single process. These threads share the processes, resources, but are able to execute independently. The threaded programming model provides developers with a useful abstraction of concurrent execution. Multithreading can also be applied to a single process to enable parallel execution on a multiprocessing system.

[3.3.1] AN IMPLEMENTATION MESSAGE PASSING APPLICATION PROGRAMMING INTERFACE WITH K-MEANS

Our main goal is to enhance the search engine performance to the maximum level it could reach. We used K-Means Text Clustering as explained to achieve the proposed system. And have achieved incredible results after more than one test and building a massive database for the operating test clustering to reduce the time taken. A challenge was found due to the shared state concurrency where all the threads in the same process space have access to the same areas of memory. We have thus decided to use a message passing concurrency with each threat that has access only to its own state in a way that helps our software as the need to synchronize memory no longer exists. The one method for a thread to communicate with other threads is by sending them messages. We have divided our system into two main parts as shown below:

- **Part 1:** In the first part we create processes to calculate the VSM through the Main Worker (MW) which initiates the workers then checks if there any processor (CPU) is available; if yes, it initiates new workers till all CPUs are used. All workers will work as many applications working concurrently to do first main function which is "Analyzing List of Documents". Each worker takes a list of documents and performs the analyzing task which includes mainly three tasks: 1- Handling distinct keywords, 2- Preparing data collection and, 3- calculating VSM. After doing the task, it examines if the data is empty and reads a second time if they were found; asking them to create a list of VSM for the list documents. as shown in **[Figure 3].**

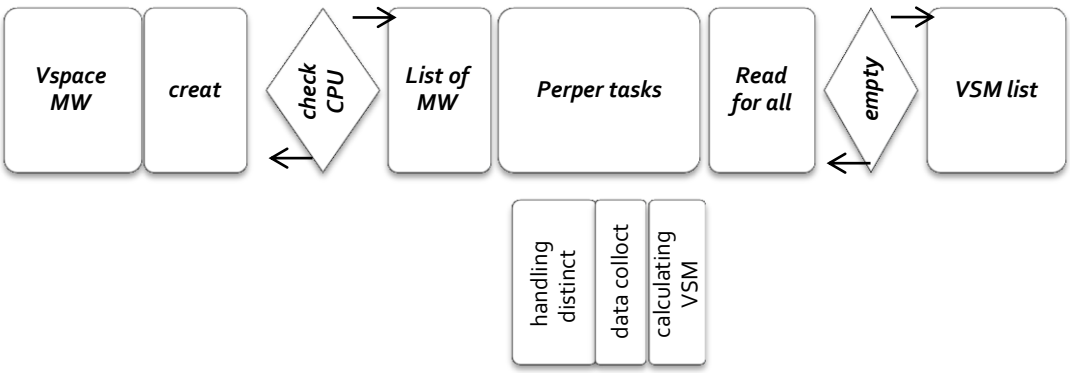

Figure: 3. Show List of VSM

- **Part 2**: In the second part, we receive a list of documents prepared and analyzed and vector space calculated for each element another list is the distinct keywords list, then this part will process these two inputs to achieve the main task of this part which is "Cluster all keywords into a number of clusters". If the first part, the main worker (MW) that initiate the workers then check if there any processor (CPU) is available if yes initiate new workers till all CPUs are used. All workers will work as many applications working concurrently to do a first main function. For each worker takes list of documents and performing the clustering task that include mainly three tasks: 1- Assigning center, 2- Iterate to create new centers and, 3- choosing the best center for each cluster. The worker then reads all the data and checks if they go back empty to give us a list of the clustering and reads again if it's not empty as shown in **[Figure 5].**

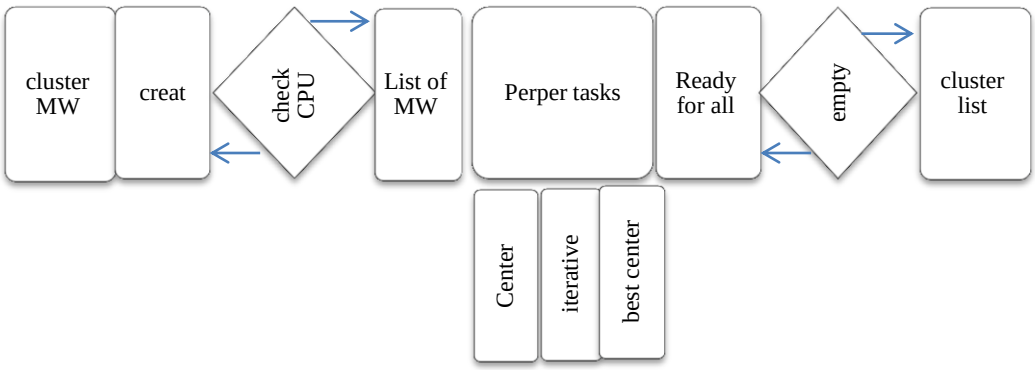

Figure :4. Show List of Clustering

[4] EXPERIMENTAL RESULTS

The accuracy and performance of the proposed MPAPI with K-Means are further verified using an experimental database sphider. All the experiments are carried out using PHP and C#. The proposed algorithms have been implemented on i5-2430M CPU @ 2.40GHz with 4GB RAM running on Microsoft Windows 7 Ultimate operating system.

The training database for the search engine sphider contains more than 47 sites and 23542 links and 400000 keywords.

The first experimental measures the efficiency of the proposed method by comparing it with other existing k-means methods (See Table 1). It reports the comparison between the two systems using the same keywords. After increasing keywords datasets, the normal K-Means clustering is becoming very slow and when dataset becomes more than 300K the whole system hung up and stopped.

To solve this problem, we did a revolutionary enhancement by integrating new K-Means algorithm with MPAPI. Experiment shows that the system can work correctly and perfectly with the sufficient training data, and cluster accuracy increases when the system is trained with more samples. As we can see MPAPI with K-Means is faster than normal K-Means because of the stack of computations performed by the proposed algorithm that doing all functionality in concurrently.

When using 'document 6' and number of keyword 100000, the K-Means take time 0.25 seconds, but MPAPI with K-Means take 0.04 seconds, and when number of document 20 and the number of keywords 200000 k-means take time 17.97 second while MPAPI with K-Means takes 0.69 seconds. Whenever increase the number of documents and number of keywords whenever k-means takes time for cluster data and hinge system, where an increase number of documents 98 and the number of keywords 400000 K-Means takes 600 seconds but MPAPI with K-Means takes 23.32 seconds. This experiment shows that our system proposes better than k-means as shown in **[Figure 5].**

Table I: Comparison of Clustering Time Between K-Means and K-Means with MPAPI.

Documents	NO. keywords	K-Means Time(Sec)	K-Means with MPAPI Time(Sec)
6	100K	0.25%	0.04%
28	200K	17.97%	0.69%
66	300K	185.98%	7.22%
98	400K	600%	23.32%

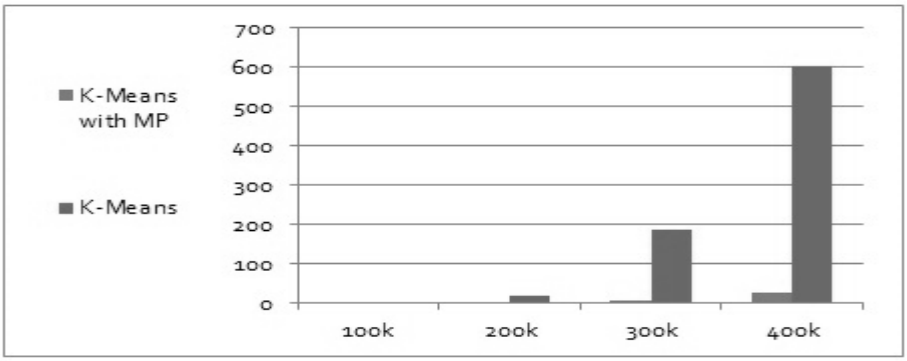

Figure: 5. Comparison of Clustering Time between K-Means and K-Means with MPAPI

In experiment 2, comparisons between the normal search engine (sphider) and search by K-Means (table 2) show when using a number of keywords (70000), when a search by normal search for word 'life' takes time 0.38 seconds, while K-Means takes (0.15) seconds, and search for word 'success' by normal search takes 0.21 seconds and k-means taking 0.15 seconds. The time difference in searching between normal and K-Means because sphider search from one database wrap 70000 keywords to finding words wanted in that need time, but K-Means clustering, database to the number of clusters then we search check cluster other cluster to find word wanted take this word and stop search rather than normal search as shown in **[figure 6].**

Table 3: Compression between Normal Search Engine (Sphider) and Search from K-Means.

Search by	Normal search Time (Sec)	MPAPI with K-means Time (Sec)
70000 k		
Life	0.38%	0.15%
Success	0.21%	0.15%
1963	0.18%	0.12%
IEEE	0.21%	0.12%
IEEE + Wikipedia	0.47%	0.25%
Frog	0.21%	0.10%
Frogs & Toads	0.40%	0.22%

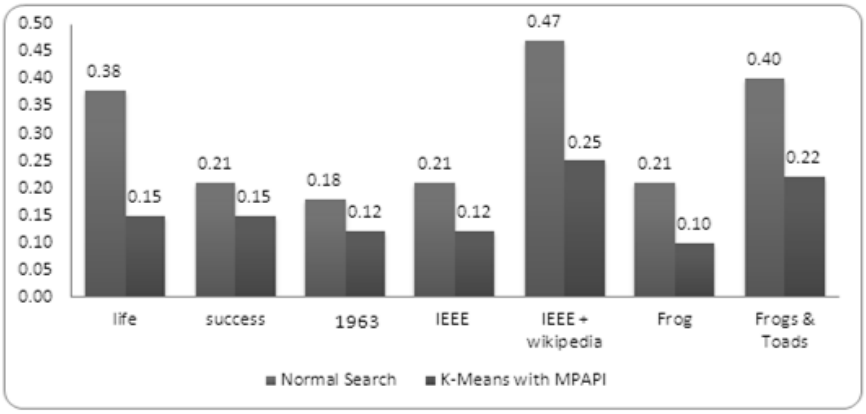

Figure: 6. Comparison between Normal Search Engines (Sphider) and Search from K-Means.

In the third experiment, we increased keywords to 400,000 words, where we got the following results when searching for the word 'life', the average time taken to search is 1.92 seconds, but search by K-Means takes 0.06 seconds and when search for word (IEEE + Wikipedia) by K-Means takes time very few 0.02 seconds reverse normal search takes time 3.59 seconds (table 3).

Table 3: Compression Between Normal Search Engine (Sphider) and Search from K-Means.

Search by	Normal search Time (Sec)	MPAPI with K-means Time (Sec)
	400000 K	
Life	1.85%	0.05%
Success	1.86%	0.01%
IEEE	1.76%	0.03%
IEEE + Wikipedia	3.59%	0.02%
Frog	1.89%	0.01%
Frogs & Toads	3.68%	0.01%
1960s-70s	1.71%	0.02%
World	2.04%	0.08%
Sakho	1.94%	0.03%

Evaluate system that all the experiences that we've had, made it clear that the proposed system K-Means with MPAPI to improve the search engine (Sphider) give a good product and is not affected by increasing the number of keywords unlike the normal search engines that the increase the number of keywords increased search time for the word one wants to search for Show in **[Figures 7 and 8].**

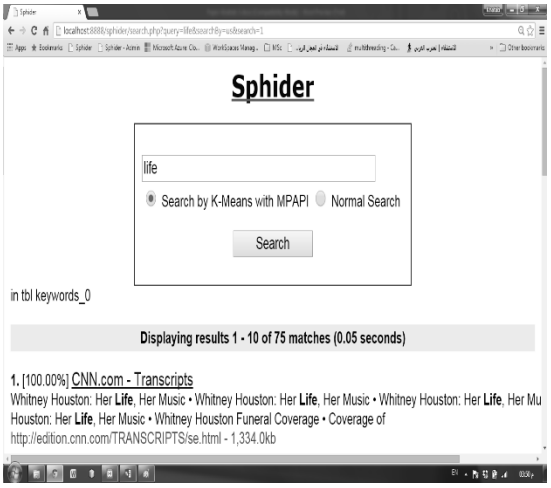

Figure :7. Shows Results from K-Means with MPAPI Search

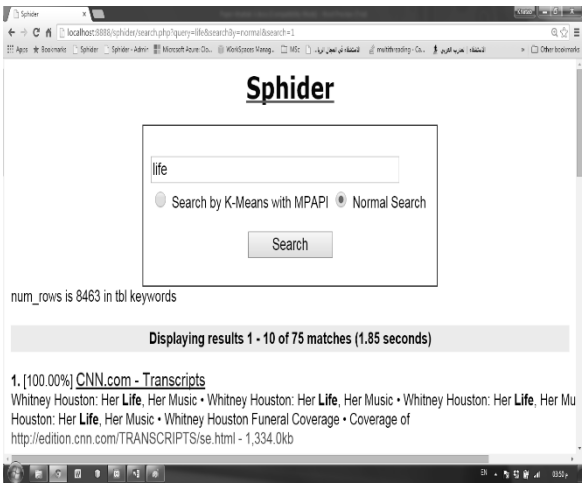

Figure: 8. Shows Results from a Normal Search

[5] CONCLUSION AND FUTURE WORK

Search engines are programs that search documents for specified keywords on a search for information on the World Wide Web and returns with a list of the documents where the keywords were found. In this research, we use the search engine Sphider It includes an automated crawler, which can follow the links found on a site, and an indexer which builds an index of all the search terms found in the pages. This research aims to improve the search time of search engines to the greatest extent using the K-Means Algorithm which does the Clustering of the database. Upon conducting experiments, it is found that the algorithm produces better results without the clustering. As the size of data increases, the time used in searching is affected as search time increases to a great extent and the search process becomes slower. We decided to use the MPAPI technique with the K-Means to solve the delay problem during search. Using this technique, search takes place in more one cluster in parallel. Each thread can connect to its own case; connection takes place through messages not State transfer between Threads. All experiences that we've made it clear that the proposed system K-Means with MPAPI to improve the search engine (Sphider) give a good product and not affected by increasing the number of keywords unlike the normal search engine that. In the future to develop more, this proposed system running the cloud computing to reduce over time the clustering keywords.

REFERENCES

[1] G. Zhang, C. Li, C. Xing, and G. Zhang, "A Semantic++ Social Search Engine Framework in the Cloud," in SKG '12: Proceedings of the Eighth International Conference on Semantics, Knowledge and Grids. IEEE Computer Society, Oct., pp. 277–278, 2012.

[2] R. Singh and S. K Gupta, "Search Engine Optimization Using Crawler Optimization- A Technique" International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 2, Issue 4, India, July – August, pp. 393-395, 2013.

[3] J. B. Killoran,"How to Use Search Engine Optimization Techniques to Increase Website Visibility", IEEE Transactions on Professional Communication, Vol. 56, no. 1, March 2013, USA, pp 50-66,2013.

[4] D. Agarwal and P. Sharma " Study on Information Retrieval Proficiencies for Mining the Network" International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 4, India, April, ISSN: 2277 128X, 2013.

[5] "Ultimate Benefits of SEO" [Online]. Available: http://www.nextsbd.com/seo/benefits-of-seo.php.

[6] M. Sharma, A. Chaudhary, M. Mathuria, and S. Chaudhary, "Review Study on the Privacy Preserving Data Mining Techniques and Approaches," International Journal of Computer Science and Telecommunications (IJCST), Vol. 4, Issue 9, pp. 162-166, September 2013.

[7] P.-N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, Addison-Wesley Companion Book Site 2006.

[8] Hongwei," A Document Clustering Algorithm for Web Search Engine Retrieval System", In proceeding of International Conference on e-Education, e-Business, e-Management and e-Learning, China, PP 383-386, 2010.

[9] K. Norvag," Authoritative K-Means for Clustering of Web Search Results", Norwegian University of Science and Technology Department of Computer and Information Science, June 2010, Norwegian, pp.16-19.

[10] D. Khurana and Dr. M.P.S bhatia," Dynamic Approach to K-Means Clustering Algorithm", international journal of computer engineering and technology (IJCET), volume 4, issue 3, may, India, pp. 204-219, 2013.

[11] PACS Training Group," Introduction to MPI", Board of Trustees of the University of Illinois, USA, pp1-202, 2001.

[12] K.K. Kattamuri and R. Chiramdasu," Search Engine with Parallel Processing and Incremental K-Means for Fast Search and Retrieval", International Journal of Advances in Engineering & Technology, Jan., India, ISSN: 2231-1963,2013.

[13] P. Jiang, C. Zhang, G.Guo, Z. Niu and D. Gao," A K-means Approach Based on Concept Hierarchical Tree for Search Results Clustering", Sixth International Conference on Fuzzy Systems and Knowledge Discovery IEEE.2009, China, DOI 10.1109/FSKD.658, pp.380-386,2009.

[14] M. Cui and S. Hu, "Search Engine Optimization Research for Website Promotion", ICM Transp. Manage., China, vol. 4, pp. 100-103,2011.