Characterizing Web Document Change

Lipyeow Lim[1], Min Wang[2], Sriram Padmanabhan[2], Jeffrey Scott Vitter[1], and Ramesh Agarwal[2]

[1] Dept. of Computer Science, Duke University, Durham, NC 27708-0129
{lipyeow, jsv}@cs.duke.edu
[2] IBM T. J. Watson Research Ctr., Hawthorne, NY 10532
{min, srp, ragarwal}@us.ibm.com

Abstract. The World Wide Web is growing and changing at an astonishing rate. For the information in the web to be useful, web information systems such as search engines have to keep up with the growth and change of the web. In this paper we study how web documents change. In particular, we study two important characteristics of web document change that are directly related to keeping web information systems up-to-date: the degree of the change and the clusteredness of the change. We analyze the evolution of web documents with respect to these two measures and discuss the implications for web information systems update.

1 Introduction

The World Wide Web is growing and changing rapidly [6]. The dynamic nature of web data poses a problem to information systems that either cache, summarize or index the web. These information systems typically have to sample or crawl the web periodically and update their local view of the web to reflect the changes in the web.

In dealing with this update problem, it is often helpful to know three characteristics of the change:

1. How *frequent* does a web document change?
2. How *much* has the content of a web document changed within a certain time interval?
3. How *clustered* are the changes in the web document?

Knowing the frequency of the change allows us to use variable crawling rates for documents with different change frequencies and thus conserve network bandwidth [4]. It also allows us to optimize for the common (most frequent) case, for example, by keeping data structures for frequently changing documents in memory [7].

Knowing how much the content of a web document has changed tells us how much the web has remained the same between two consecutive crawlings (samplings). Knowing the distribution of these changes tells us whether the changes are spread out in the changed document or whether these changes are

well clustered and thus only affect small portions of each document. If changes are large and well spread out, rebuilding the local view of the web from scratch every time the web is crawled (sampled) may be more efficient than an incremental approach. On the other hand, if changes are small and clustered, an incremental approach may be more efficient.

The frequency of the web document change has been studied in previous work [5, 3, 2, 1, 4]. In [3], Cho et al. discuss how the frequency of change can be modeled by a Poisson process and how the frequency of change can be estimated from observed data. They also discuss the implications of these frequency estimates on crawling the web in [4]. Brewington et al. removed the memoryless assumption implicit in a Poisson process by modeling the web changes as a Renewal process [2, 1]. They further defined a *freshness* metric to characterize how up-to-date a local information repository is compared to the web. Douglas et al. analyzed web changes using web access traces that yield distributions of web documents with respect to a variety of metrics [5]. However, because of the access-driven nature of their method, their results may not reflect the less popular documents on the web.

Despite the importance of the two other characteristics of the web change, no serious study has been done previously on these two issues. In this paper we address these two important questions: how much has a typical web document changed during two consecutive crawlings and how clustered are the changes.

We define two measures, a distance measure and a clusteredness measure, for analyzing and quantifying web document change. The distance measure characterizes the size of the change between two versions of a web document and the clusteredness measure characterizes how these changes are spread out within a web document.

The rest of the paper is organized as follows: In the next section, we have a general discussion on the types of changes between two samples of the web. In Section 3 we describe the data set that we used in our distribution analysis. In Section 4 and Section 5 we define the two measures for web document change and present our data analysis results with respect to these two measures. We discuss the implications of our data analysis on web information systems in Section 6 and conclude in Section 7

2 Types of Web Document Changes

In this section, we describe the types of changes that occur between two consecutive crawlings of the web. The set of web documents obtained from one crawling at a particular time is called a*sample* of the web and the time between two consecutive samples is called the *sampling interval*. We define a web document to be the sequence of words contained in a HTML file that has been stripped of scripting code and HTML syntax. Each HTML file or each document is associated with an URL and is assigned a unique document ID (doc_id). Each word occurrence within a document encodes the information $\langle word\_id, doc\_id, loc\_id \rangle$, which is also known as a *posting*, where $word\_id$ denotes the the unique ID iden-

tifying each word in the (English) vocabulary and *loc_id* denotes the position of that word occurrence in the document. Since each posting encodes all the information in a word occurrence on the web, the entire web can be encoded as a set of postings and a web information system can be viewed as a system maintaining the set of all postings (or a subset of it). For example, a web index is a web information system that stores this set of postings ($\langle word\_id, doc\_id, loc\_id \rangle$) sorted by *word_id*.

If we consider each sample as a set of documents, then between two consecutive samples S_n and S_{n+1}, any document can only belong to one of the following partitions (see Figure 1 top diagram),

$$\begin{cases} S_n \cap S_{n+1} & \text{common documents} \\ S_n - (S_n \cap S_{n+1}) & \text{deleted documents} \\ S_{n+1} - (S_n \cap S_{n+1}) & \text{inserted documents.} \end{cases} \quad (1)$$

In the Venn diagram at the top of Figure 1, a point represents a document, so

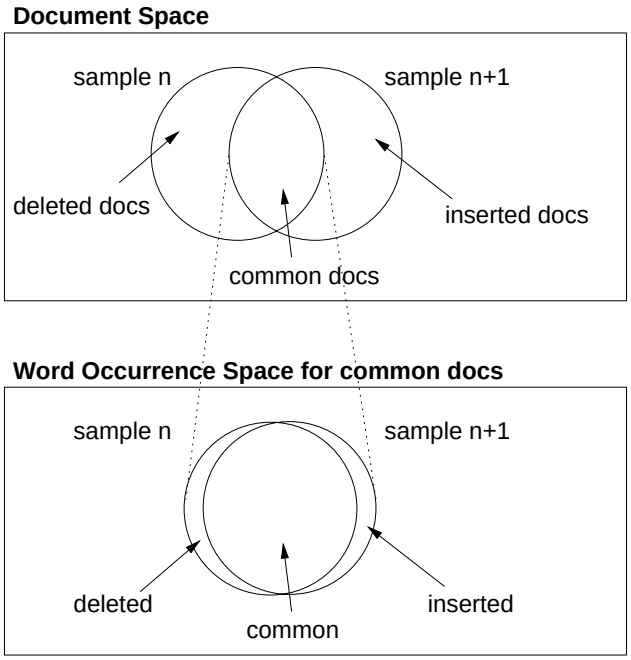


Fig. 1. Types of changes at the document collection level and the word occurrence collection level.

each circle represents a set of documents. Deleted documents are old documents that do not exist in the new sample anymore and they need to be removed from the system. Inserted documents are new documents that appears in the new

sample only and they need to be inserted into the system. Common documents are the documents that are common between the two consecutive crawlings of the web. Some of these common documents might be unchanged and some of it might contain changes.

In this paper we are primarily concerned with the changes in the common documents. We have three reasons for this emphasis. The first is that techniques for inserting documents into information systems have already been well studied by the text retrieval community. Secondly, the common documents represents at least 50% of the documents currently maintained by the web information system at each update. Finally, of these common documents, most of the changes at each update are small as we will see in Section 4. If we consider the set of common documents as a collection of word occurrences (defined next), then between two consecutive samples, the overlap is significantly large.

We can further examine the set of common documents at the granularity of a word occurrence. A word occurrence corresponds loosely to a posting (a $\langle word\_id, doc\_id, loc\_id \rangle$ tuple) without the limitation of a numeric location ID; that is, it is shift invariant in some sense. In the Venn diagram at the bottom of Figure 1, each point represents a word occurrence. Each circle represents the set of word occurrences corresponding to the set of common documents of a particular sample. If we consider each circle as a sequence of word occurrences, the set of common word occurrences loosely corresponds to the longest common subsequence of the two sequences. Since most web information systems keep track of word occurrences, the number of common word occurrences gives an upper bound on the number of postings in the system that can remain unchanged upon an update.

3 Data Set Description

For our analysis we recursively crawled web documents starting from several seed URLs up to a maximum recursion depth of five levels. The list of seed URLs consists of www.cnn.com, www.ebay.com, www.yahoo.com, espn.go.com, and www.duke.edu. Our sampling interval is 12 hours (at 7 am and 7 pm EST). We collected data over a period of one month. For the data that we present here, we use 2 samples that are representative of the general update behavior.

Each document is preprocessed into a canonical form by stripping off any HTML tags and scripting code. Each character is transformed to its uppercase and extra white spaces are stripped.

Other characteristics of our data are summarized in Table 1. Note that although we perform our analysis over many samples of the web over many days, we only present the data analysis for a set of representative data since this is a clearer presentation than using 3D plots.

No. of docs at time n	6042
No. of docs at time $n+1$	6248
No. of deleted docs	2788
No. of inserted docs	2994
No. of common docs	3254
No. of common docs unchanged	1940

Table 1. Summary of the representative data set used for data analysis.

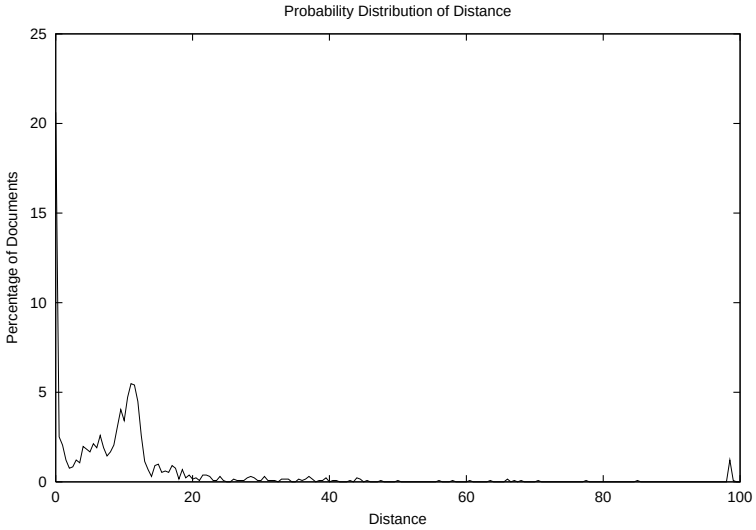
4 Degree of Change

We quantify the degree of the change between two versions of a document with a distance measure. We define our distance measure based on the idea of *edit distance*. Edit distance is usually defined as the minimum number of *edit operations* (insertions or deletions) required to transform one sequence to another. A document can be considered as a sequence of words or *word_id*'s and hence the ideas of edit distance map naturally to document distance as well. We define δ to be the minimal number of words deleted or inserted to transform one document to another. Our distance measure for two documents A and B can then be defined as

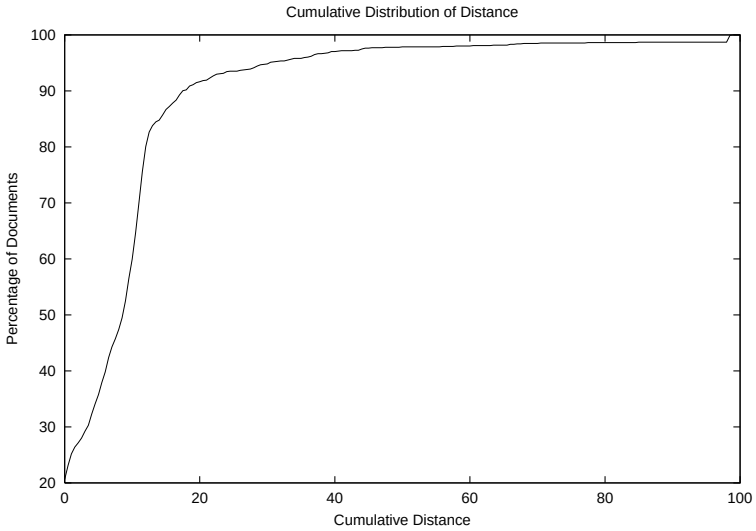
$$d(A,B) = \frac{\delta}{m+n}, \tag{2}$$

where m and n are the size (in words) of document A and document B respectively. Clearly, δ can be computed once the longest common subsequence of the two documents is known. If the two documents are the same, δ will be zero and the distance measure will be zero. If the two documents are completely different, δ will be equal to $m+n$ (since m old words need to be deleted and n new words need to be inserted) and the distance measure will be one.

We use this distance measure to obtain the distribution of the common documents that have changed with respect to the magnitude of change (Figure 2). The data points in our plots are computed by obtaining the distance of every old and new document pair and classifying them into bins where each bin corresponds to an interval of 0.5%. Each data point therefore corresponds to the number of documents in a particular bin. From the probability distribution plot with respect to our distance measure (Figure 2), we observe that most documents fall into the bins between distance 0 % and 20 %. From the corresponding cumulative distribution plot, we see that more than 90% of the documents have changes smaller than a distance of 20%. Moreover this behavior seems consistent across updates, i.e., over time, and it shows that the set of common word occurrences is very big and a large portion of the information maintained by the web information system can remain unchanged at an update.



(a) Probability Distribution



(b) Cumulative Distribution

Fig. 2. Distribution of documents with respect to our distance measure.

5 Clusteredness of Change

Besides the magnitude of change we are also interested in how clustered the changes are. For example, the insertion of a paragraph of 10 words to the beginning of a document is surely different from inserting the same 10 words at 10 different random and non-contiguous locations in the document. Why should this be of concern? Suppose the document is stored as an array of *word_id*s. The first type of change requires shifting all the words after the insertion point once by 10 cells. The latter type of change at random locations requires 10 separate shift operations. A similar argument can be constructed for linked-list or tree representation of the document; hence how localized a change is does affect the amount of computations required.

How do we measure clusteredness? One possibility is to use a clustering algorithm to find the position (the center) of each cluster and calculate the distance of each change from the nearest cluster (similar to the idea of statistical variance); however, this means that our clusteredness measure will be dependent on how good the clustering algorithm is and that is not desirable. Instead we resort to a simpler but effective method. We choose a block size and partition the document into blocks according to the size we have chosen. The fraction of the blocks affected by changes can therefore be used as a measure of how clustered the change is. The clusteredness of the changes required to transform document A to document B is

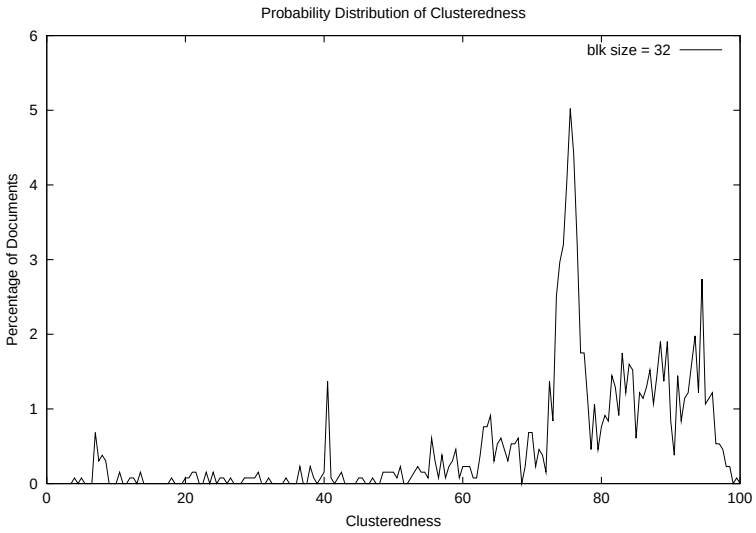
$$c(A, B, b) = 1 - \frac{\Delta}{\lceil m/b \rceil}, \tag{3}$$

where Δ is the number of blocks affected by the change, m is the size of the old document in words and b is the block size in words. If there are no changes, Δ will be zero and the clusteredness will measure one. If all the changes are clustered into one block and assuming that the block size b is sufficiently small, the clusteredness will be close to one. If the changes are distributed over all the blocks, Δ will be equal to $\lceil m/b \rceil$ and clusteredness will be zero.

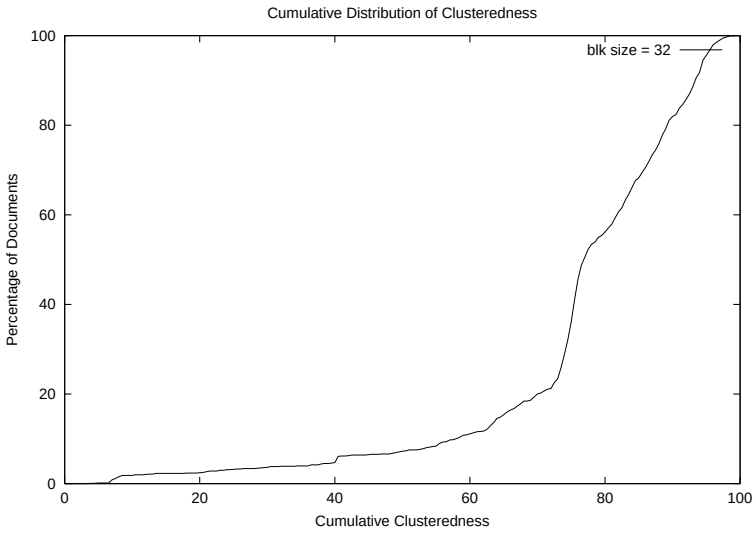
The number of blocks affected by the change, Δ, can be determined in practice by first finding the (minimal) edit transcript between the old and the new document. An edit transcript is the sequence of edit operations required to transform the old document to the new document. In UNIX, the output of the `diff` command is a representation of the minimal edit transcript between two files. Using the edit transcript we can determine where the edit operations occur and count the number of blocks affected by the edit operations.

The block size b must be chosen such that it is much smaller than the document size m for this measure to be meaningful. Another possibility is to partition the document using HTML tags. One such tag is the paragraph tag `<p>`.

We study the distribution of the documents using this clusteredness measure. Since the clusteredness measure is only meaningful for documents that have changed, we perform our data analysis only on those documents. Two partitioning schemes are used: fixed size blocks (Figure 3) and `<p>`-tag blocks (Figure 4). From the probability distribution plot with respect to our clusteredness measure



(a) Probability Distribution



(b) Cumulative Distribution

Fig. 3. Distribution of documents with respect to clusteredness $c(A, B, 32)$.

$c(A, B, 32)$ (Figure 3), we observe that most document have changes that are more than 50% clustered, that is the changes affect less than half of the blocks. From the corresponding cumulative distribution plot, we see that only about 20% of the documents have changes that are less than 70% clustered. Using HTML paragraph tags, we observe in the probability distribution plot with respect to the clusteredness measure $c(A, B,$ <p>-tag) (Figure 4) that many document have changes that are more than 50 % clustered; however significant spikes occur consistently at the 0-0.5 % clusteredness bin. This is because some documents do not use the <p>-tag at all. For such documents there is only one block in total and any changes must occur in that block and hence $c(A, B,$ <p>-tag) is zero. This is consistent with the previous $c(A, B, 32)$-distribution plot (Figure 3) where no documents have changes distributed to every block. Other observable artifacts are the spikes at the 50% and 66% clusteredness marks. These are mostly due to the documents with only two to three paragraphs in total.

We observe that these plots show a very skewed distribution of documents across the clusteredness measure.

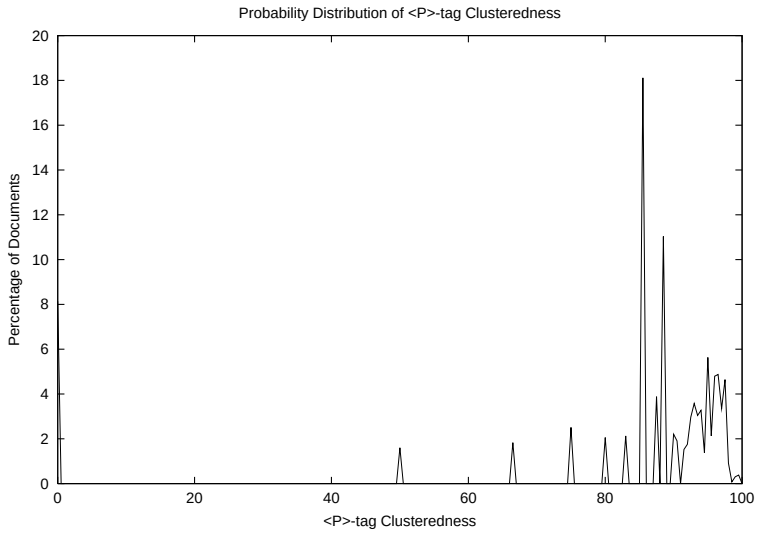
6 Implications for Web Information Systems

We show in Section 4 and Section 5 that the most changes in web documents are small and clustered. These skewed distributions expose an opportunity to improve the performance of web information systems by optimizing for the frequent case. In this section, we describe how we can exploit such distributions with a simple example. The example is chosen to illustrate the principles involved rather than for realism. For a non-trivial example, we refer the reader to our upcoming paper which will show how these techniques can be applied to updating web indexes.

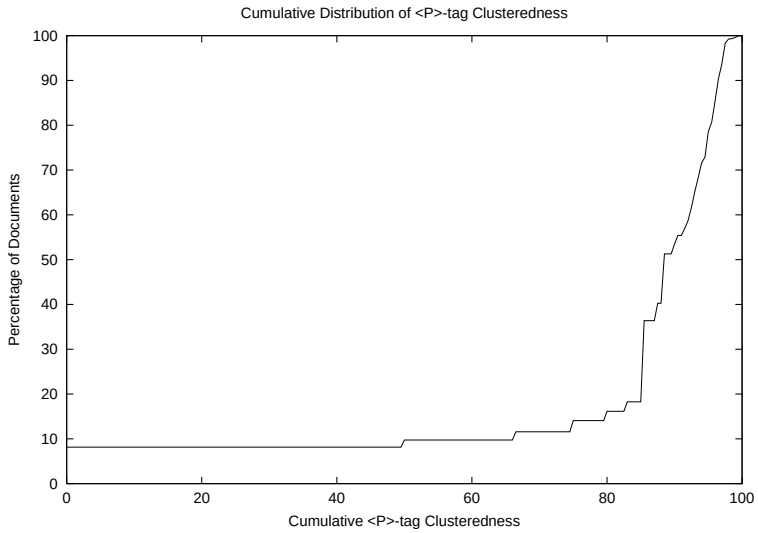
Suppose we want to cache the entire web on a local file system. Our clusteredness results suggest that if each document is stored as several files each of size 32 words, then every update touches only a relatively small number of files. This means that we can leave a large portion of files untouched.

Using the same web caching example, suppose we would like to count the number of words that need to be touched during an update. A word occurrence is said to be *touched* if it is deleted or inserted or shifted in position. Further suppose that within a block/file that needs to be modified, only the words occurring after the starting position of the first change in the block need to be touched (since words occurring before the first change in the block remains unchanged). How does block/file size affect the number of words that need to be touched?

We measured the minimum number of words touched for different block sizes (using the same data set as the previous data analysis) and verified that as the block size gets smaller the number of edit operations decreases. It should be clear that if the block size is one word, this measure becomes the distance measure we defined in Section 4. Figure 5 shows the distribution plots of the documents with respect to the number of words touched normalized by the size of the old and the new document in number of words.

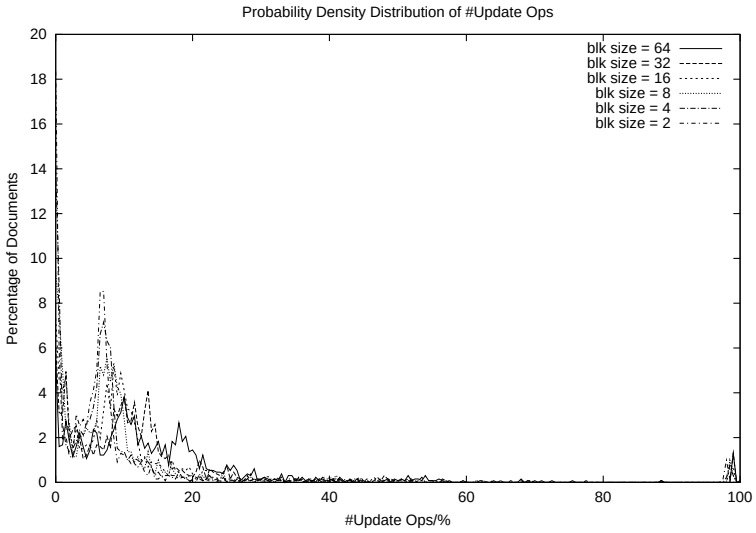


(a) Probability Distribution

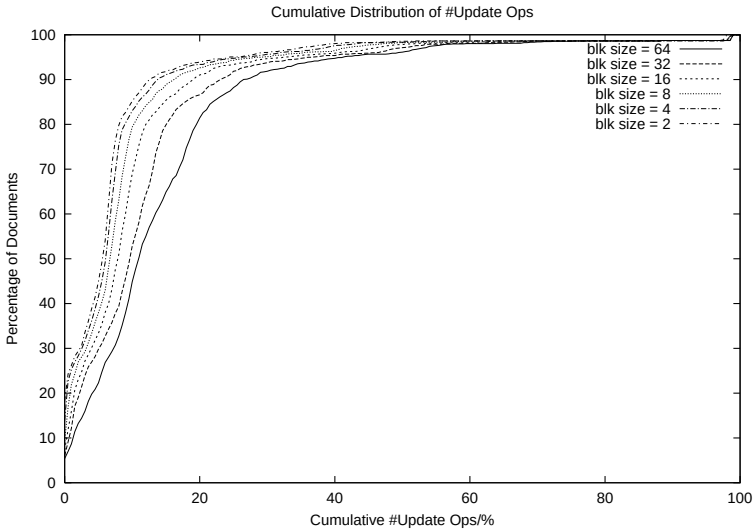


(b) Cumulative Distribution

Fig. 4. Distribution of documents with respect to clusteredness $c(A, B, \texttt{<p>-tag})$.



(a) Probability Distribution



(b) Cumulative Distribution

Fig. 5. Distribution of documents with respect to the number of edit operations for different block sizes.

Note that in our caching example, there is a trade-off between the block/file size (hence number of files) and the size of the tables maintained by the file system. Extremely small block sizes should be avoided, because as block size gets smaller, the size of each of the tables maintained by the file system grows larger.

In practice, it is possible to determine the optimal block size by first finding the average cost of a file system table lookup relative to touching a word and then determining the block size that gives the optimal distribution for the two types of operation.

7 Conclusion

The dynamic nature of web data poses an challenging problem to web information systems on its efficient maintenance. In this paper we defined two measures, a distance measure and a clusteredness measure, to quantify some aspects of the dynamism of the web data so that we can have a basis on efficient maintenance of web information systems. Our analysis of the web document changes using these two measures have shown that we document changes are generally small and clustered, suggesting that update methods based on an incremental approach can be much more efficient compared with naive methods that need to rescan all the web data.

References

1. B. Brewington and G. Cybenko. How dynamic is the web? In *Proceedings of the Ninth International World Wide Web Conference*, May 2000.
2. B. Brewington and G. Cybenko. Keeping up with the changing web. *IEEE Computer*, 33(5):52–58, May 2000.
3. J. Cho and H. Garcia-Molina. Estimating frequency of change. *Submitted for publication*, 2000.
4. J. Cho and H. Garcia-Molina. The evolution of the web and implications for an incremental crawler. *26th International Conference on Very Large Data Bases*, September 2000.
5. F. Douglis, A. Feldmann, B. Krishnamurthy, and J. Mogul. Rate of change and other metrics: A live study of the world wide web. *Proceedings of the USENIX Symposium on Internet and Systems*, 1997.
6. S. Lawrence and C. L. Giles. Accessibility of information on the web. *Nature*, 400:107–109, 1999.
7. A. Tomasic, H. Garcia-Molina, and K. Shoens. Incremental updates of inverted lists for text document retrieval. *Proceedings of 1994 ACM International Conference of Management of Data (SIGMOD)*, pages 289–300, May 1994.