Concerning Etags and Datestamps

Lars R. Clausen

The State and University Library[1]
Universitetsparken
8000 Århus C
Denmark
Email: lc@statsbiblioteket.dk
Phone: +45 89 46 22 53

In web archiving, avoiding unnecessary downloads of unchanged pages can significantly reduce the load on both the archiving system and the server being archived. However, the indicators available for determining whether a page is changed are frequently either missing or wrong, causing pages changes to missed. In this paper, we investigate the quality of the two change indicators defined in the HTTP protocol, Last-Modified and Etag. Based on downloads of front pages of Danish web sites, we compare the reliability and usefulness of the two indicators and consider if using a combination of the two can lead to better prediction of page changes. Finally, we present a systematic way to determine the best prediction scheme, and present an unexpected download scheme with better characteristics than the obvious choices.

Keywords: HTTP Web archiving Change prediction Datestamps Etags

1. Introduction

When archiving web pages, one quickly notices that a number of pages are the same for a considerable amount of time, while others change content frequently. In order to avoid wasteful downloading and archiving of many copies of the same content, we are interested in finding a scheme for reliably predicting whether content has changed without having to download the content. Reducing the amount of downloads will reduce the load not only on the web archive's server, but also on the server being archived.

HTTP, the protocol used for downloading web pages, gives various meta-data when a web page is downloaded. In this paper, we examine prediction schemes that are solely based on the two indicators for change of content in the HTTP headers: The

[1] This paper was prepared in cooperation with The Royal Library in Copenhagen as part of the netarchive.dk project

datestamp (Last-Modified) and the *Etag*[C]. The datestamp is the last time the content has been modified, assuming the server can determine that. The Etag (Entity Tag) header is part of the cache control mechanism. A caching proxy or browser can compare the Etag of a cached copy of a page with the one received in the headers for a new page, and may assume that if the Etag is the same, the content is the same. Thus, an unchanged Etag should indicate that the previous copy can be used.

In theory, use of the datestamp and Etag should allow download of exactly those pages that have changed, and only downloading the headers of other pages. However, in practice many servers send out no change indicators or change indicators that are not consistent with changes in the actual content, as has been documented in several reports[D][E][G][F]. Based on downloads of Danish web pages, we will investigate how using the datestamp or Etag or combinations thereof allows us to predict changes in the content.

We will first consider four schemes for deciding when to re-download a page, given an old version of the same page:

Scheme 1: Download when the datestamp is missing in either page or changed between the two pages

Scheme 2: Download when the Etag is missing in either page or changed between the two pages

Scheme 3: When both datestamp and Etag are present in both pages, download if they have *both changed*, otherwise download if neither is present or if the one indicator present indicates change.

Scheme 4: When both datestamp and Etag are present in both pages, download if *either of them has changed*, otherwise download if neither is present or if the one indicator present indicates change.

The latter two try to combine the results of the two indicators to get a higher-quality result, and differ only in their prediction when both indicators are present. These four schemes together represent the four obvious ways we can use the change in indicators to predict whether a page has actually changed. In section 4, we will see how systematic assessment of possible schemes can produce non-obvious but useful schemes.

Avoiding download of the entire body can be done in three different ways: By using a HEAD request first, and then GET if the content is predicted changed, by using a GET and breaking connection after receiving the header if the content is

predicted unchanged, or by using the If-None-Match and If-Modified-Since headers in the request. The first method involves an extra connection for each download, which takes time and server resources. The second method requires low-level interaction with the web client, which may be complex to implement. The third method is preferable, though it is subject to errors if the web server does not implement the required functionality correctly, and it may not be able to support more complex prediction schemes.

This report investigates how well the four schemes described above predict change in a sampling of front pages of Danish web sites, whether the quality of the indicators is randomly distributed or depends on the server, and whether other schemes exists that work better than the four proposed above.

1.1 Terminology

To be of use for avoiding duplicate download, a scheme must be *reliable* and *useful*. Reliability means that when the content changes, the scheme must predict the change, or we will miss an update. Reliability is measured in the percentage of changed pages we decide to download. The perfectly reliable scheme downloads 100% of pages that have changed. The trivial scheme of always downloading a page is also 100% reliable, but gives no performance benefit.

Usefulness means that when the content does not change, the scheme should predict the lack of change, or we will download unnecessarily. Usefulness is measured in the percentage of unchanged pages we avoid downloading. A perfectly useful scheme skips the download of 100% of unchanged pages, i.e., it never downloads a page that has not changed. The trivial scheme of never downloading any pages is 100% useful, but of course 0% reliable, while always downloading is 0% useful, as no unnecessary downloads are avoided.

2. Methodology

In our experiment, we sampled the front pages of all Danish second-level domains every other night over a period of one month. For each page, we recorded the date, the Etag, the size, and an MD5 sum of the body of the page.

The list of Danish domains was generously supplied by DK Hostmaster, and contains 465,374 domains in the `.dk` top-level domain. We harvested only the front page of each domain by requesting http://www.<domain>.dk/. We used Wget version 1.8.2 with the parameters –S –r –l 0 –t 1 –T 30[2]. The contents of each page were

[2] -S: Write HTTP headers, -r: recursive, -l 0: 0 recursion levels, -t 1: 1 retry, -T 30: IP timeout 30 seconds

stored in a file and an MD5 checksum of the body was found using md5sum. The headers, size of body and MD5 sum of body were stored for further processing.

After each harvest, the resulting files were processed with a Perl script to extract the name of the domain, the size, the MD5 sum, the Etag header, and the Last-Modified header. In case of redirects, only the data for the last page in the redirect chain was used, but stored under the name of the domain. Missing headers were marked in the file as well.

Once the harvesting had been done, the processed data was compared to find changes in Etags, datestamps and MD5 sums. We checked each domain for changes between successful downloads, and for each such download we recorded whether the contents (not the headers) had changed since the last successful download, whether datestamps and Etags were present, and whether or not they had changed. Some servers sent out Etags or datestamps on some but not all visits. For purposes of content change prediction, a change between having Etags/datestamps and not having them was considered the same as if two different Etags/datestamps were given.

3. Results

We performed 16 harvests of the front pages of the Danish web domains. On average, 346,526 web servers were contacted in each harvest, the number varying from 340,937 to 351,585. The total number of different servers contacted was 361,408, the remaining servers were inaccessible or did not have a working HTTP server at any of 16 times we attempted connection. A total of 5,543,470 entries were found when processing the downloaded data, with an average body size of 3,897 bytes.

Since we want to look at changes in content, we will look at the downloads where the page had been downloaded in a previous harvest (*consecutive downloads*). Table 1 presents the number of consecutive downloads found, as well as how many of them had changed content. As can be seen, over 80% of the downloads done in this experiment could have avoided if an accurate predictor of content changes had been available.

Total pages	Total consecutive downloads	Total changed	Total unchanged
5,543,470	5,182,034	599,143 (11.6%)	4,582,891 (88.4%)

Table 1: Number of pages, consecutive downloads and changes in the consecutive downloads

In table 2, we show how many datestamps and Etags were found in the downloaded pages, and how well they, where found, predict whether content has changed. Not all servers send out Etags or even datestamps. Of the pages downloaded, only 3,321,598 (59.9%) had Etags. Datestamps are much more common, with 5,539,430 pages (99.93%) having a datestamp. Both indicators are fairly reliable,

missing less than 1% of changed pages, the Etag missing less than 0.1%. They are somewhat conservative, and predict changes in up to one-third of pages that have not changed.

	Exists in pages	Exists in consecutive downloads	Mispredicts change	Mispredicts non-change
Datestamp	5,539,430 (99.93%)	5,178,421 (99.93%)	1,780 (0.30%)	1,659,866 (36.2%)
Etag	3,321,598 (59.9%)	3,123,939 (60.28%)	520 (0.087%)	553,905 (12.1%)

Table 2: Frequency and quality of the datestamp and Etags in downloaded pages

To evaluate the reliability and usefulness of our proposed schemes, we must consider how accurate their predictions would have been when including the pages without Etags or datestamps. We do this by checking, for each scheme, respectively how many pages would have been downloaded out of the 599,143 that changed content, and how many pages would have been downloaded out of the 4,582,891 avoidable downloads. The results of this are shown in table 3.

Scheme	Changes missed	Reliability	Unnecessary downloads	Usefulness
1 (date)	1780	99.70%	1,662,579	63.7%
2 (Etag)	520	99.91%	2,179,045	52.5%
3 (date and Etag)	2026	99.66%	1,645,670	64.1%
4 (date or Etag)	1706	99.72%	2,132,044	53.5%
Always download	0	100%	4,582,891	0%

Table 3: Reliability and usefulness of the four schemes when missing indicators are taken into account.

Since a scheme can only predict non-change if the indicators for it are present, the reliability for schemes 1 and 2 are the same as the mispredictions of change in table 2, but the usability includes the pages missing indicators, and so is not as good as the mispredictions of non-change. When the missing indicators are considered, using Etags (scheme 2) actually gives the least savings in number of pages downloaded. Using just the date (scheme 1) gives a lower reliability, but higher usefulness.

The two hybrid schemes fall in between, with scheme 3 (download when all available indicators have changed) being slightly less reliable and slightly more useful than scheme 1, and scheme 4 (download when at least one of the available indicators has changed) combining the poorer reliability of scheme 1 with the lower usefulness of scheme 2. Schemes 3 and 4 are less reliable than scheme 2 because pages without Etags are downloaded based on the datestamp alone. All schemes give a reliability of over 99.5% and a usability of over 50%. If 99.9% reliability is not required, schemes 1 and 3 are the best, avoiding almost two-thirds of unnecessary downloads. We shall

see in section 4 that better schemes can be found through a systematic evaluation of all possible schemes.

3.1 Quality of existing Etags

It is worth noting that datestamps are significantly more common than Etags. We could attempt to educate webmasters about the importance of good Etags, both to us in terms of lesser storage requirements and to them in terms of less bandwidth usage. Let us for a moment consider the hypothetical situation where all servers send both date and Etag for all pages. We might simulate the reliability and usefulness this would entail by looking at just the pages that have both datestamps and Etags. There are 3,116,927 such consecutive downloads, with changes occurring in 92,916 of them (3.0%).

Scheme	Changes missed	Reliability	Unnecessary downloads	Usefulness
1 (date)	348	99.79%	101,349	96.6%
2 (Etag)	520	99.69%	553,905	81.3%
3 (date and Etag)	594	99.64%	84,440	97.2%
4 (date or Etag)	274	99.84%	570,814	80.7%

Table 4: Reliability and usefulness of the four schemes when ignoring Etag-less downloads.

As we can see in table 4, the usefulness of Etags increases significantly when discarding the downloads without Etags, as would be expected. More surprisingly, the usefulness of the datestamp increases even more. The number of needlessly downloaded pages when using Etags fell by a factor four, but fell by more than an order of magnitude when using datestamps. Scheme 3 downloads a mere 1/35th of needless pages here. However, since almost all the pages not considered here were missing Etags, scheme 2 fares much worse in reliability – all its errors are retained for a smaller dataset. In fact, both scheme 1 and 4 are now more reliable than scheme 2, as their reliability is almost unchanged.

This result does not mean that if everybody started using Etags, Etags would necessarily become less reliable (though it could happen). The fact that the usefulness of the datestamp is so high when missing Etags entries are removed indicates that those servers that implement Etags are also more careful to deliver a correct datestamp. This led us to wonder whether there is a correlation between server types and quality of datestamps, but a cursory examination of the server types and datestamp quality shows no indication of any such correlation.

It is also worth noting that while 65% of the overall consecutive downloads had both Etags and datestamp, only 28% of the changed pages have both. This difference probably skews the results, as only 3% of the downloads considered in this section have changes. There could be a number of explanations for this result, for instance that frequently changing pages tend not to have Etags, but we have not examined this difference in depth.

The results for this hypothetical situation should be taken with a grain of salt. Apart from it being unlikely that all servers start implementing Etags properly, the correlations between Etags quality and datestamp quality found here might not persist if more servers were to implement Etags, as new implementations might be of lower quality, especially if done by individual web masters.

3.2 Server assessment

To improve reliability, we may consider whether unreliable Etags and datestamps are sent consistently by a few servers or occur randomly across a number of servers. If the unreliable indicators are isolated on certain servers, we could treat those differently from the majority, and thus improve reliability without severely affecting usefulness.

A total of 5,987 servers sent out Etags with some but not all pages. A total of 145,755 servers sent no Etags whatsoever, but closer examination of the contents of the Server header line does not reveal any correlation between the server type and whether Etags were sent. This indicates that the decision whether or not to send Etags is made by the webmasters rather than by the server developers. Thus, any educational attempts at increasing the number of Etags being sent out should be aimed at webmasters rather than server developers, although better server support would help.

14 servers sent new content at every download without changing the Etag at all. These account for 210 out of 414 cases of unreliable Etags, over 50%. Similarly, new content was sent out at every download without changing the date by 65 servers, accounting for 975 (58%) of the errors for datestamps. If a downloading system could notice this fact and download their pages every time regardless of Etags and datestamps, we would get a reliability for Etags of 99.95% and for datestamps of 99.85%, about twice as good as without such a system.

It is noticeable that of the 14 servers that consistently sent unreliable Etags, 9 were hosted by Geocities and featured a hit counter as the only change, and the remaining 5 were from www.m.dk, which features a continuously updating status field. Of the 65 servers that consistently sent unreliable datestamps, 46 contained a randomly generated session identifier, which is different for each download, but does not affect the content. It is likely that the majority of missed changes are actually such identifiers.

The converse problem, of servers sending new Etags or datestamps without changing content, cannot be addressed in the same way. We could notice that some servers have this problem, but if we chose to not download their pages, we would miss any updates that might happen. However, in scheme 4, where we choose to download if either of the indicators have changed, knowing that one indicator updates too often on a particular server could allow us to disregard that indicator for that

server. Preliminary experiments with this idea show little promise, possibly because the servers that have both indicators tend to implement both with the same quality. Thus, the case where one indicator is useful but the other is updated too often seems to be infrequent.

4 Systematic assessment of schemes

The four schemes chosen are the four reasonable schemes if the two indicators are independent. However, as was seen in section 3.2, there is a strong correlation between the presence and accuracy of the Etags and of the datestamps. Thus, schemes that exploit this correlation rather than seeing the indicators as independent boolean variables may work better. To investigate this possibility, we devised a way to automatically describe all possible schemes based on the two indicators.

Any scheme that decides whether to download based on whether the Etag and datestamp indicators present have changed can be described by what they predict for each entry of a 3x3 matrix as shown in table 5.

	Etag changed	Etag unchanged	Etag missing
Datestamp changed			
Datestamp unchanged			
Datestamp missing			

Table 5: The decision matrix for a download scheme.

For each entry, a scheme must decide whether to download or not. From our data, we can fill in how many pages falling into each category had changed, and how many had not. In table 6, we show the filled-in the matrix with the numbers in each entry indicating how many pages of that category had changed, respectively had not changed.

	Etag changed	Etag unchanged	Etag missing
Datestamp changed	165,594/84,440	246/16,909	430,623/1,558,517
Datestamp unchanged	74/469,465	274/2,386,937	1,432/63,910
Datestamp missing	0/0	0/0	900/2,713

Table 6: The decision matrix with amounts of changed and unchanged pages in each entry.

Since there is 9 categories to make choices for, we can exhaustively describe the possible schemes by what their choices are in each category, leading to a total of 512 schemes. To assess how good a scheme is, we sum the changed pages in the categories where the scheme downloads to get the reliability in percent of the total number of changed pages, and sum the unchanged pages in the categories where the

scheme does not download to get the usefulness in percent of the total number of unchanged pages.

Out of these schemes, the majority are of very low quality and can immediately be discarded. When considering only those that have a reliability of over 99% and usefulness over 50%, we get 64 schemes (though because two of the entries in the matrix are empty, these come in groups of four that have the same results). Out of these schemes, those whose reliability and usefulness are both bested by some other scheme can be removed. This leaves us with 28 schemes, of which one from each group of 4 with the same results is shown in table 7[3].

Download if Etag is:	Same	Same	Same	Changed	Changed	Changed	Missing	Missing	Missing		
and Datestamp is:	Same	Changed	Missing	Same	Changed	Missing	Same	Changed	Missing	Reliability	Usefulness
IgnoresMissing1					X			X		99.51%	64.15%
Scheme 3					X	X		X	X	99.66%	64.09%
Scheme 1		X	X		X	X		X	X	99.70%	63.72%
IgnoresMissing2					X		X	X		99.75%	62.76%
Scheme 3 variant					X		X	X	X	99.90%	62.70%
Scheme 1 variant		X			X	X	X	X	X	99.94%	62.33%
Scheme 4 variant		X		X	X		X	X	X	99.95%	52.08%

Table 7: The 7 schemes with the best reliability and usability.

Of these 7 schemes, Scheme 1 and Scheme 3 from earlier turn up as the second and third most useful, with the most useful being only 0.1% more useful than Scheme 1 and somewhat less reliable.

Three other schemes are variants of schemes 1, 3 and 4 from above, but with the difference that a page is always downloaded when the Etag is missing. These three variants are particularly interesting. The Scheme 1 variant has a reliability of 99.94% and a usefulness of 62.33%, combining the best points of Scheme 1 and Scheme 2. The Scheme 3 variation is slightly more useful but somewhat less reliable, while the Scheme 4 variation is slightly more reliable but significantly less useful. The two schemes named IgnoresMissing are both schemes that avoid downloading when both indicators are missing. Such schemes would deal poorly with reduced numbers of

[3] The schemes shown were selected for being the ones where the selection criteria are most easily described.

datestamps and should be avoided. They only appear here because so few pages lack datestamps.

An interesting way of looking at the Scheme 1 variation is that it essentially says that when the Etag header is missing, the datestamp is totally unreliable, and we should always download, but otherwise the datestamp is the best indicator. This matches the findings shown in section 3.1, where the datestamp showed a significant improvement when the Etag-less pages were not considered. Note that this scheme cannot be implemented using the If-None-Match and If-Modified-Since headers, as they do not implement a strong enough logic for the server to decide whether to send the body.

The new scheme found here is the best match for our data, but might not be the best for situations with different characteristics. However, the approach used can be applied to any set of data to determine the prediction scheme that works best in that situation.

5. Related Work

Little work has been done to quantify how useable datestamps and Etags are in a web archiving context. Some web archiving papers mention the poor quality and lack of presence of datestamps[I][L], and several suggest using previous change frequency[L][K] or page importance ordering[J] to decide when to crawl a page again.

The web caching community has done more intensive studies of the quality of datestamps and Etags, since a cache only has those few pieces of information available to decide whether to use a cached copy or not. However, their focus is not on the average behavior of all web pages, but on the behavior of the pages that are requested by the users. User requests frequently center on either long-term stable sites (e.g. reference works) or continuously updating sites (e.g. news listings)[F].

Wills and Mikhailov have examined cacheability of web pages in two papers, one based on harvesting specific sites[D], and one based on monitoring the user requests passing through a cache[G]. The former is the most reminiscent of our situation, in that they use popularity ratings from 100hot.com to select sets of web pages that they subsequently harvest. They find fewer and lower-quality datestamps and Etags than we, but find a similar relative distribution of quality. As they do not consider possible correlations between datestamps and Etags, we cannot tell whether our prediction schemes would be applicable for their data.

Wills and Mikhailov also note that the most popular web server, Apache, generates Etags based on the datestamp by default, and can generate randomly changing Etags due to the use of inodes in the Etag[M]. Newer versions of Apache allows the generation of the Etag to be specified by the user, and webmasters are advised to not

use inodes for Etag generation. Dynamically generated web pages normally do not have Etags nor datestamps, but can be coded to generate them based on e.g. MD5 checksums (see for instance [H]).

Mogul has also investigated the quality of datestamps for the purpose of using them to run a cache[E]. He found 71% of responses had datestamps, with 3% of them being unreliable, but does not consider the Etag headers.

The fact that the other investigations of datestamps and Etags show much lower quality than ours can indicate either that our dataset is unusual (being only front pages), that their datasets are unusual (being selected by popularity), or that significant improvements in datestamp and Etag quality has occurred in the four years that have passed since the earlier investigations. Cursory examination of other harvests show no indication that the datestamp and Etags are more prevalent now, but do not answer the question of whether the data presented in this article are representative of a comprehensive harvest. Further investigation using a proper cross-section of accessible pages would be needed to answer this question. In any case, the systematic assessment of prediction schemes can still be used to determine the best scheme, both for web archives and for cache systems.

6. Conclusion

Web crawlers face the problem of having to download a large number of pages, many of which do not change from crawl to crawl, and thus needlessly increase the resources required for crawling. We have examined whether the Etag and Last-Modified HTTP headers can be used to predict whether a page has changed. We examined 16 consecutive harvests of the front pages of all Danish internet domains with a total of 5,182,034 consecutive downloads of pages. We compared changes in the content to changes in the Etag and Last-Modified headers.

We have presented a method for determining the best prediction scheme, based on comparisons of datestamps, Etags and MD5 sums of the content. This method allows us to find the scheme that gives the best result for a particular purpose, and has shown that the best schemes are not necessarily the obvious ones.

We have found that it is possible to predict changes accurately enough that less than 0.1% of changes are missed while over 60% of unnecessary downloads are avoided. We cannot reduce the unnecessary downloads by more than 65% without missing over 25% of changes. The best strategy always downloads when the Etag header is missing, and otherwise downloads only when the Last-Modified header indicates change or is missing. This yields 99.94% accuracy of predicting change and 63.3% accuracy of predicting non-change.

The Etag header has been shown to be more reliable but less useful than the Last-Modified header. Using the Etag header to decide whether to re-download a web page

would erroneously omit only 0.087% of all changed pages, and would avoid downloading unchanged pages 52.5% of the time. The Last-Modified header would give errors in 0.30% of all changed pages, but would avoid 63.7% of unnecessary downloads. The most obvious hybrid methods yield reliability and usefulness comparable to using just one of the indicators, but a systematic examination of possible schemes yielded methods that combine the best of the two indicators.

While the Last-Modified is present almost universally, the Etag header is missing in 40% of all downloads. When looking at only the pages containing Etag headers, the Last-Modified header becomes a better predictor of content change than Etag, being slightly more reliable and only downloading less than 5% of unchanged web pages. This may simply indicate that servers that send out Etag headers are more careful about sending correct datestamps. The missing Etag headers are not attributable to particular server software, suggesting that to get better Etag coverage, webmasters need to be educate about the value of quality Etags.

A few servers consistently send out new content without changing the Etag or Last-Modified headers. If the pages from such servers are always downloaded regardless of the headers, the number of missed changes can be halved without severely affecting the number of unnecessary downloads. A similar scheme does not appear viable for servers sending out new headers for unchanged content.

Regardless of which scheme is used, use of the If-None-Match and If-Modified-Since headers should yield better performance than downloading the headers separately to check the indicators and can reduce the number of downloads by between one-half and two-thirds without missing significantly many changes. However, the quality of implementations of these headers remains to be investigated.

One possible source for systematic error is the fact that we only look at front pages, not the rest of the site. While one would assume that the existence and quality of the header fields would be the same on any one server, the rate of change and average size of pages might be different in deeper pages. Additionally, the distribution of number of pages per server follows a power law, so the majority of pages in an archive would come from relatively few servers. An additional study of the quality of Etags and datestamps based on random selection from a large archive would be interesting future research.

Acknowledgements

The author would like to thank in particular Niels H. Christiansen of The Royal Library, Denmark, for comments on the article and for suggesting the idea of systematic assessment. Thomas Zäschke of the State and University Library, Denmark, Xavier Roche, maintainer of the HTTrack web crawler, Mikhail Mikhailov of Worcester Polytechnic Institute, MA, and the anonymous reviewers also provided useful comments and critique of the article.

References

A. Arvidson, "Kulturarw3", in *Proc. Preserving the Present for the Future*, Copenhagen 2001, pp.101-104.

B. P. _abi_ka, "Archiving the Czech Web: Issues and Challenges", in *Proceedings of 3rd Workshop on Web Archives,* Trondheim 2003, pp.111-117.

C. R. Fielding et al., "Hypertext Transport Protocol – HTTP/1.1", RFC 2616.

D. Craig E. Wills and Mikhail Mikhailov. Examining the cacheability of user-requested web resources. In *Proceedings of the 4th International Web Caching Workshop*, San Diego, CA, March/April 1999.

E. Jeffrey C. Mogul. *Errors in timestamp-based HTTP header values.* Research Report 99/3, Compaq Computer Corporation Western Research Laboratory, December, 1999.

F. Fred Douglis, Anja Feldmann, Balachander Krishnamurthy, and Jeffrey Mogul. Rate of change and other metrics: A live study of the World Wide Web. In *Proc. USENIX Symp. on Internet Technologies and Systems*, pages 147--158, December 1997.

G. Craig E. Wills and Mikhail Mikhailov. *Towards a better understanding of web resources and server responses for improved caching*, Technical Report WPI-CS-TR-98-27, Computer Science Department, Worcester Polytechnic Institute, December 1998.

H. Mark Nottingham. *cgi_buffer*, URL: http://www.mnot.net/cgi_buffer/

I. Daniel Gomes and Mário J. Silva. *A characterization of the Portuguese Web*, 3rd ECDL Workshop on Web Archives, Trondheim, Norway, 21 August 2003

J. S. Abiteboul, G. Cobena, J. Masanes, G. Sedrati. A First Experience in Archiving the French Web. In: Maristella Agosti, Costantino Thanos (eds): *Proceedings of the ECDL 2002*. Rome, Italy (September 2002). Springer - Lecture Notes in Computer Science.

K. Junghoo Cho, Hector Garcia-Molina: *Estimating Frequency of Change*. Research Paper, Stanford, 2000

L. Brian E. Brewington and George Cybenko. How dynamic is the web? *WWW9 / Computer Networks*, 33(16) :257-276, 2000.

M. The Apache Software Foundation. *Apache Core Features,* 2004, URL: http://httpd.apache.org/docs-2.0/en/mod/core.html