Emerald Insight

Internet Research

Search engines crawling process optimization: a webserver approach

Mhamed Zineddine

Article information:

To cite this document:
Mhamed Zineddine , (2016),"Search engines crawling process optimization: a webserver approach", Internet Research, Vol. 26 Iss 1 pp. 311 - 331
Permanent link to this document:
http://dx.doi.org/10.1108/IntR-02-2014-0045

Downloaded on: 09 November 2016, At: 20:29 (PT)
References: this document contains references to 57 other documents.
To copy this document: permissions@emeraldinsight.com
The fulltext of this document has been downloaded 305 times since 2016*

Users who downloaded this article also downloaded:

(2016),"Estimating Google's search engine ranking function from a search engine optimization perspective", Online Information Review, Vol. 40 Iss 2 pp. 239-255 http://dx.doi.org/10.1108/OIR-04-2015-0112

(2007),"Analysing Google rankings through search engine optimization data", Internet Research, Vol. 17 Iss 1 pp. 21-37 http://dx.doi.org/10.1108/10662240710730470

Access to this document was granted through an Emerald subscription provided by emerald-srm:563821 []

For Authors

If you would like to write for this, or any other Emerald publication, then please use our Emerald for Authors service information about how to choose which publication to write for and submission guidelines are available for all. Please visit www.emeraldinsight.com/authors for more information.

About Emerald www.emeraldinsight.com

Emerald is a global publisher linking research and practice to the benefit of society. The company manages a portfolio of more than 290 journals and over 2,350 books and book series volumes, as well as providing an extensive range of online products and additional customer resources and services.

Emerald is both COUNTER 4 and TRANSFER compliant. The organization is a partner of the Committee on Publication Ethics (COPE) and also works with Portico and the LOCKSS initiative for digital archive preservation.

*Related content and download information correct at time of download.

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

The current issue and full text archive of this journal is available on Emerald Insight at:
www.emeraldinsight.com/1066-2243.htm

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

Search engines crawling process optimization: a webserver approach

Mhamed Zineddine
Management Information Systems, ALHOSN University, Abu Dhabi, UAE

Received 8 February 2014
Revised 16 February 2014
20 February 2014
6 March 2014
14 December 2014
6 February 2015
Accepted 8 February 2015

Abstract

Purpose – The purpose of this paper is to decrease the traffic created by search engines' crawlers and solve the deep web problem using an innovative approach.

Design/methodology/approach – A new algorithm was formulated based on best existing algorithms to optimize the existing traffic caused by web crawlers, which is approximately 40 percent of all networking traffic. The crux of this approach is that web servers monitor and log changes and communicate them as an XML file to search engines. The XML file includes the information necessary to generate refreshed pages from existing ones and reference new pages that need to be crawled. Furthermore, the XML file is compressed to decrease its size to the minimum required.

Findings – The results of this study have shown that the traffic caused by search engines' crawlers might be reduced on average by 84 percent when it comes to text content. However, binary content faces many challenges and new algorithms have to be developed to overcome these issues. The proposed approach will certainly mitigate the deep web issue. The XML files for each domain used by search engines might be used by web browsers to refresh their cache and therefore help reduce the traffic generated by normal users. This reduces users' perceived latency and improves response time to http requests.

Research limitations/implications – The study sheds light on the deficiencies and weaknesses of the algorithms monitoring changes and generating binary files. However, a substantial decrease of traffic is achieved for text-based web content.

Practical implications – The findings of this research can be adopted by web server software and browsers' developers and search engine companies to reduce the internet traffic caused by crawlers and cut costs.

Originality/value – The exponential growth of web content and other internet-based services such as cloud computing, and social networks has been causing contention on available bandwidth of the internet network. This research provides a much needed approach to keeping traffic in check.

Keywords Information retrieval, World wide web, Search engines, Crawlers, Deep web, Networking traffic optimization

Paper type Research paper

1. Introduction

Information communications technology has become an essential part of our daily activities. The number of internet users has been increasing faster than ever. It is estimated that internet users reached 46.4 percent globally in 2015 (IWS, 2015). The wide range of internet services such as e-mail; word wide web (WWW); file transfer; voice over internet protocol (IP); instant messaging; chat; peer to peer content sharing software; high definition (HD) video (Onural *et al.*, 2006) and voice streaming; and electro-holography (Holovideo) (Niwase *et al.*, 2013) have led to the generation of massive data traffic that has to be supported by the internet infrastructure. The current internet was designed to be resilient. However, the bandwidth required to exchange content generated by companies and users (Web 2.0 and later 3.0) will bring it to its knees, if nothing is done to mitigate the issue (Laudon and Traver, 2008). Many fast gigabit networks have been developed in the world to deal with this issue; however, most of these networks are still limited to research and educational institutions.

Internet Research
Vol. 26 No. 1, 2016
pp. 311-331
© Emerald Group Publishing Limited
1066-2243
DOI 10.1108/IntR-02-2014-0045

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

A colossal amount of new content is generated every day. This content has to be stored, accessed, exchanged, or searched by users. To facilitate accessibility and reachability of available content, it has to be categorized and indexed to be searchable using keywords submitted by users. Search engines offer a critical service that enables users to search available content quickly and effectively (Laudon and Traver, 2008). To be indexed and categorized, search engines have to collect and analyze this content, and subsequently its updates. They use crawlers or spiders to fetch content, which lead to the generation of extra traffic between search engines and web servers. This paper is dealing with the optimization of the traffic created by search engines, when collecting new and updated content using a new approach.

The body of the paper is organized as follows: Section 2 presents the background of this study; Section 3 presents the framework of this research; Section 4 exhibits existing methods; Section 5 discusses the proposed approach; Section 6 presents the experiment; Section 7 presents results and analysis; Section 8 discusses the results; and Section 9 concludes the paper.

2. Background

2.1 Online content growth

The internet and its services have become a necessity in our daily activities, whether it is communication, e-business, e-commerce, social media, or other types of content dissemination and retrieval activities. The qualities of the internet include being ubiquitous, providing global reach, using universal standards, enabling information richness, information density, and others. These characteristics facilitated an exponential growth of web content and servers hosting it (Laudon and Traver, 2008). The web was estimated to host more than 100 billion documents (Argaez, n.d.) and it is increasing at a speedy rate. Many offline applications such as gaming and office applications have been migrating to the internet. The concept of software as a service provided through the cloud is the main driver for this trend. Content which is broadcasted nowadays might be unicasted online for the masses in the near future such as TV, three-dimensional TV, and other services (Brodkin, 2012). The issue is that there is no single choking point to measure the internet total traffic. Cisco has been publishing figures of IP-based data crossing the global network. In 2012, global IP traffic was estimated to be 43.6 exabytes per month and forecasted to reach 120.6 exabytes per month by 2017 (Cisco, 2013). Efforts on many facets have been directed to mitigate traffic issues. Upgrading the infrastructure of the internet is one facet, optimizing communication between systems and networking traffic is another. Scaling the infrastructure is not always the right recourse; using the existing one effectively may be more appropriate in many cases. OpenFlow, for instance, is software designed to help mitigate bandwidth issues. It has been adopted by Google and being examined by Worldwide Large Hadron Collider Computing Grid (Cisco, 2013).

The problem with the exponential growth of content has been the lack of quality control on the content generated. Due to the massive size of the content generated on the WWW, quality content has become scarce and locating it has become a challenging task. Hence, a more effective way to locate quality content from the web was required, and search engines emerged from such a need.

2.2 Search engines

Search engines are special web-based systems enabling access to web content. They use crawlers (i.e. spiders or sleuths) to collect information using preset criteria. Content collected is classified and indexed for fast search and access (Eijk, 2009). Search engines

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

are extensively used by web surfers to locate relevant content according to keywords submitted (Table I, Figure 1). Google uses what is called Page Rank Software, which measures the importance or popularity of each page, solving an equation with more that 500 million variables and two billion terms to determine the best pages for the query (Laudon and Traver, 2008). The database used to look up relevant content or what may be categorized as best web content matching the query, is populated using sophisticated programs. Crawlers or spiders were designed to collect web content in order to be analyzed, categorized, and indexed. Crawlers start with a list of seed URLs and branch out by extracting URLs from the pages visited. Web crawlers navigate a directed graph based on a breadth-first algorithm until all links are visited or the crawlers have run out of resources. The dilemma is that computing resources are finite and the web content on the internet is very dynamic and grows explosively. The best practices suggest that website owners should keep their sites up to date in order to promote search engines' crawling. In addition to the new content generated every day, it was estimated that 52 percent of web content changes every day (Cho and Molina, 2000). In order for search engines to keep their indexes up to date, crawlers have to recursively revisit web servers and download updated content. The problem is that crawlers have to decide what web servers to revisit and when. Much work has been done to estimate change frequency (Winkler, 1972; Misra and Sorenson, 1975; Cho and Molina, 2000; Taylor and Karlin, 1998; Press release, ComScore, 2014), however, crawlers had been found to generate approximately 40 percent of internet traffic (Yuan and Harms, 2002). Moreover, Incapsula, a web security firm reported that web traffic generated by bots reached approximately 61.5 percent in 2013, up by 21 percent from 2012 (Zeifman, 2013). Despite search engines' efforts, they have been facing serious challenges due to explosive growth and updates' dynamics (Risvik and Michelsen, 2002; Ke *et al.*, 2006), therefore, retrieving all web content has been impossible (Liu and Du, 2014).

2.3 Deep web and crawling issues

The structure and distribution of web content are complex. Inaccessible web content has emerged from such complexity. According to Bowtie Theory, 80 percent of the content of the web (deep web) may not be visible (Gopinah, 2005). The Bowtie model suggests that the web is mainly divided into five parts: a strongly connected component named SCC; a component named IN which includes web pages that can reach the SCC but cannot be reached from the SCC; another component named OUT, which includes web pages that are accessible from the SCC but do not link back to it; TENDRILS, which include pages that can neither reach the SCC nor be reached from the SCC; and the totally disconnected component named DISC (Broder *et al.*, 2000; Yuan and Harms, 2002; Gopinah, 2005). Deep web crawling is a challenging task. Massive Deep Web Databases

	Explicit core search share (%)		
Core search entity	March 12 (%)	April 12 (%)	Point change
Total explicit core search	100.00	100.00	N/A
Google sites	66.40	66.50	0.1
Microsoft sites	15.30	15.40	0.1
Yahoo! sites	13.70	13.50	−0.2
Ask network	3.00	3.00	0
AOL, Inc.	1.60	1.60	0

Table I. comScore explicit core search share, April 2012 vs March 2012. Total U.S. – home and work locations adapted from comScore qSearch (8)

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

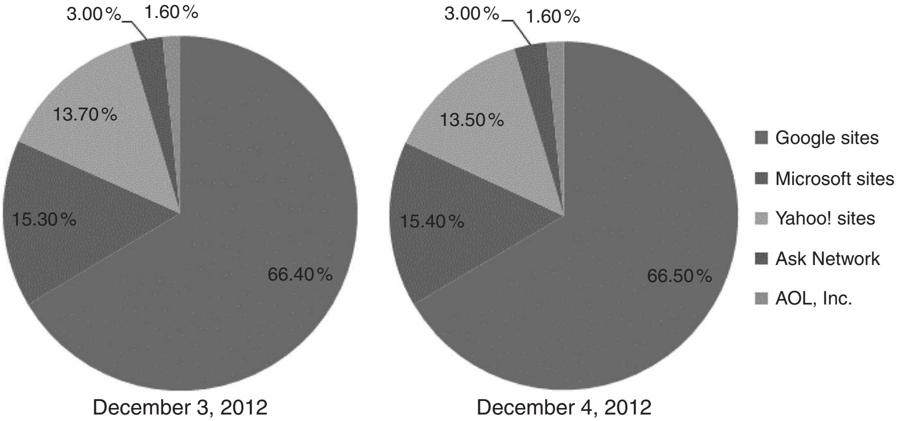

Figure 1.
Explicit core search share, April 2012 vs March 2012

(WDBs) have been hosting rich and high-quality information that is hard to retrieve, integrate, and index. Effective identification of WDBs' entry points has been problematic. However, solutions have been proposed to mitigate this problem such as Enhanced Form-Focused Crawler for domain-specific WDBs (Li *et al.*, 2013) and other algorithms directed toward enabling crawlers to learn and optimize their strategies (Zheng *et al.*, 2013). Database-driven websites and dynamic content generated using inputs from forms and other input mechanisms that require human interaction add to these issues and make much of web content hidden from crawlers. Even though many researchers have tried to propose techniques to effectively crawl hidden web (Myllymaki, 2002; Peisu *et al.*, 2008; Li *et al.*, 2013; Zheng *et al.*, 2013), the content indexed by search engines is by no means equal to all the data and information available on the internet. In the late 1990's, individual search engines indexed about 16 percent of all information on the internet and combined indexed no more than 42 percent of available web content (Lawrence and Giles, 1999). Nowadays, Web 2.0 and digital technology evolution have made web content size volatile. Indexing all generated web content has become close to impossible. Available statistics are questionable due to difficulties facing researchers in estimating exact web content at a point of time. However, many algorithms and techniques such as extrapolation, uniform sampling methods, and capture-recapture methodology might shed some light on the real size of surfaced web and deep web content (Bergman, 2001; Anagnostopoulos and Stavropoulos, 2011).

Sullivan (2012) stated that Google search engine has seen 30 trillion URLs online, which requires 100 million gigabytes to index. Moreover, on average Google spiders crawl 20 billion URL per day. The breakdown of web pages by httparchive.org suggests that the average size of text (scripts, style sheets, and HTML) in a web page is 422 kilobytes (kb), which is about 22 percent of the size of the page. Thus, the total content crawled by Google alone can be roughly estimated to be at least 8.44 petabyte (pb). The complexities facing crawlers when collecting visible and hidden content are amplifying the problem of indexing live web pages on the internet.

2.4 Content type diversity and crawling issues

Webmasters are required to optimize their websites for search engines ranking purposes. Multiple successful techniques used by search engines optimizers have been used to promote page rankings in Google (Evans, 2007) and other main search engines.

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

The issue nowadays is the diverse types and structure of web content available on the internet. Forums, social bookmarking sites (SBS), online blogs (OB), online social networks are being populated with considerable web content every day, which needs to be indexed. Their structure, however, has forced search engines to develop configurable crawlers to mitigate their crawling difficulties. Moreover, the amount and quality of the data collected are affected by crawlers' settings (Perez-Sola and Herrera-Joancomarti, 2013). Therefore, each type of these websites should be crawled by configured spiders behaving according to its structure. This observation led to the development of Forum Crawler Under Supervision (FoCUS), a "supervised web-scale forum crawler." FoCUS simplifies the forum crawling issue to a URL-type identification one (Jiang *et al.*, 2013). The same crawling challenge is caused by OB, where information retrieval and extraction have been a hindrance for search engines' crawlers. RetriBlog, which is an architecture-centered framework, seems to mitigate this issue (Ferreira *et al.*, 2013). SBS are used to store users' relevant bookmarks, and are complicated to crawl. Deciphering relevant URLs that need to be crawled from irrelevant data (i.e. noise) has been a challenging task for normal crawlers. Focussed crawling of tagged web resources using ontology was designed to alleviate this issue (Punam *et al.*, 2013).

Beside specific crawling techniques, algorithms, configurations, and methods, the concept of focussed crawling has been the center of many studies. Focussed crawlers use past crawling information to assess the relevance of new links. However, the performance depends on the type of models used and the quality of past observations. Two probabilistic models for focussed crawling – maximum entropy Markov model and linear-chain conditional random field have been proposed to deal with this issue (Liu and Milios, 2012). Furthermore, a decentralized learning automata-based focussed web crawler has been proposed to alleviate similar crawling dilemmas (Torkestani, 2012). Cho and Molina suggested incremental crawling under different conditions. They stressed that an incremental crawler should be designed to keep the data gathered fresh and improve its quality (Cho and Molina, 2012). In addition, Sharma *et al.* (2003a, b) proposed tagging web content to differentiate between volatile content and static content using a file with a TVI extension. Furthermore, Singhal *et al.* (2010) proposed a new approach to regulate the revisiting frequency, a new mechanism and architecture for the incremental crawler. Madaan *et al.* (2010) also proposed a new architecture to continuously update the hidden web depositary. Moreover, others focussed on parallel crawler processing by combining augmentations to hypertext documents (Sharma *et al.*, 2003a, b, 2010).

Less research has focussed on the web server, where the original content resides. Different search engines have different capabilities and techniques to collect, store, and index web content. Thus, their shares of explicit core search (Table I, Figure 1) differ and their web content overlap is limited (Spink *et al.*, 2006), which limits content coverage by a single search engine. Further, the diversity of web content, its structure, and web systems serving it have been a serious challenge to crawlers. Search engines crawl sites and generate traffic that otherwise would be avoided. In addition, a huge amount of content, called "Deep Web" is still unreachable (Gopinah, 2005). Web systems have to be fully engaged in the crawling process. The proposed approach in this study should mitigate some of these issues, mainly the coverage issue leading to deep web and the traffic created by crawlers. Web servers will push only necessary content (updates) to keep web content fresh, when search engines process, classify, and index it accordingly.

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

3. Research objective of the study

The research objective of the present study is to propose a novel approach aimed at optimizing the traffic generated by search engines' crawlers and extend the scope of available web content coverage by search engines; thus mitigating deep web issue. Earlier research focussed on crawlers' optimizations techniques. The presented approach focusses on the web server, which is the source of the content to be indexed and categorized by search engines. The effectiveness of the approach will be tested by measuring traffic generated using existing methods and traffic generated when the proposed approach is implemented.

4. Framework of the study

As shown in Figure 2, bandwidth and web content are increasing somewhat at the same pace. To keep the difference Δ under control, efforts were directed to keep the explosion of web content under control and to improve the bandwidth available. Δ is defined by the following equation:

$$\Delta = \beta - (\gamma + \xi) \tag{1}$$

where β is the minimum bandwidth required to handle transmitted digital content, γ the maximum amount of web content that needs to be handled and transmitted by the internet, and ε is the size of other digital content.

The aim of this study is to help keep Δ under control, and subsequently γ.

5. Existing methods

Search engines are creating excessive traffic between web servers and search engines databases. However, major search engines such as Google, Yahoo, and Microsoft have been collaborating to ease the burden on webmasters by providing standards such as Sitemap, Robots Exclusion Protocol (REP) and other enhancements including auto-discovery and cross-host submission (Garg, 2008).

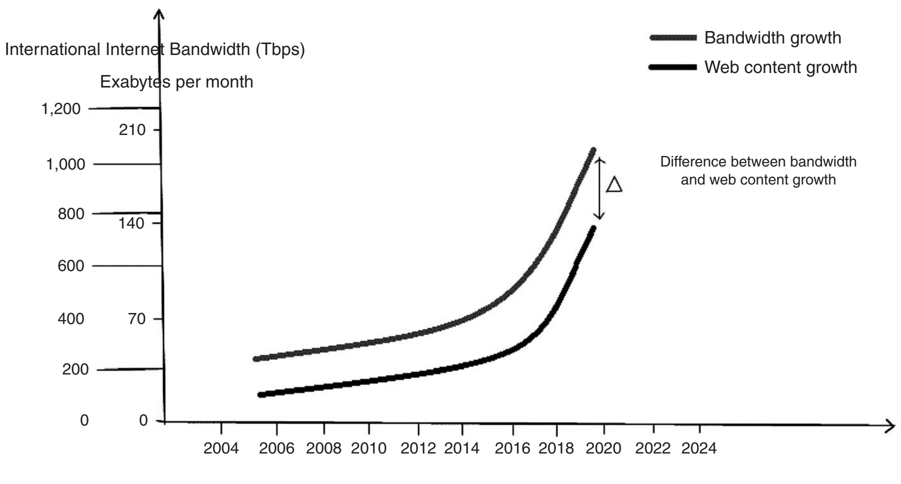

Figure 2. The evolution of the bandwidth and web content (inspired by (32, 33))

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

5.1 REP

REP was launched in 1994 (Koster, 1994). Its flexibility and simplicity promoted its adoption and implementation by major search engines (Garg, 2008). REP helped contents publishers to indicate which parts of their sites are public and which parts are private. The protocol offers two ways to control the visibility of the content *vis-à-vis* the crawler. Webmasters have been using a file called robot.txt which relates to the whole site or META tags at the page level (Garg, 2008). Directives such as Allow, Disallow, $ Wildcard Support, and Sitemap Location can be used to direct the crawler. In addition, HTML META tags directives such as NOINDEX META tag, NOFOLLOW META tag, NOODP META tag, NOARCHIVE META tag, and NOSNIPPET META tag are used at the page level to instruct the crawler.

5.2 Sitemaps

Sitemaps are an efficient way to guide search engines in the crawling process. Sitemap is an XML file that describes when a URL was last updated, how often it usually changes, and its level of importance *vis-à-vis* other URLs using metadata. The use of Sitemaps enables crawlers to crawl the site intelligently. The current version of Sitemap protocol is 0.90. It is widely adopted by search engines such as Google, Yahoo!, and Microsoft (Sitemaps.org).

Sitemap example:

```
< ?xml version = “1.0” encoding = “UTF-8”? >
< urlset xmlns = “www.sitemaps.org/schemas/sitemap/0.9” >
< url >
< loc > www.example.com/ < /loc >
< lastmod > 2005-01-01 < /lastmod >
< changefreq > monthly < /changefreq >
< priority > 0.8 < /priority >
< volatiletag >  vol < /volatiletag >
< /url >
< /urlset >
```

Sitemaps and REP help protect websites from being crawled blindly by enabling webmasters to decide what needs to be crawled and what should not. However, the problem is that the process is more manual than automated.

6. Other enhancement techniques

To help crawlers with efficiency, a focussed crawler applying the cell-like membrane computing optimization algorithm (CMCFC) was used to optimize object corresponding weighted factors and subsequently minimize the root measure square error of priorities of hyperlinks. CMCFC could be used to guide crawlers to collect higher quality web pages (Liu and Du, 2014). A focussed crawling system based on semantic ranking was proposed to guide web crawlers retrieving relevant web content (Du *et al.*, 2013). Focussed crawling was also discussed by Uemura *et al.* (2012). Further, many researchers have focussed on mobile agents to improve crawling. Index-based change detection technique and distributed indexing using mobile agents is one example (Badawi *et al.*, 2013). Other studies propose architectural designs to enhance crawling (Yan *et al.*, 2002; Yalçin and Köse, 2010). However, the involvement of web server software has been overlooked.

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

7. Proposed approach

This paper proposes a new approach combining REP, Sitemaps, and web services or any other communication mechanism where the web server is a major player in the crawling process. Web servers are more aware than search engines of the content added or modified in the websites/domains they host. Web servers are the place where changes and updates are implemented; thus, controlling, monitoring, and reporting these changes seem to be a logical part of their functions. The responsibility of suggesting what needs to be crawled and updated should be granted to the web server software. Limited manual intervention should be given to webmasters for administration purposes. A checking mechanism should be implemented by the search engine in order to prevent any misuse of the system for ranking purposes. The Figure 3 below shows a typical architecture of a search engine and Figure 4 shows the proposed one.

7.1 Pull method

The web server automatically monitors changes on each domain and updates the existing Sitemaps' XML files for each domain or for all the domains hosted by the same server. Web servers will report to the search engine any changes that have to be made using REP for managing visibility, and web services or other communication services for delivery. The search engine picks and queues the XML file for further processing. Sitemaps' files should integrate the function offered by robot.txt for ease of use. In summary, the pull method provides the search engine with the necessary information in the form of a map to be used when crawling what needs to and is permitted to be indexed.

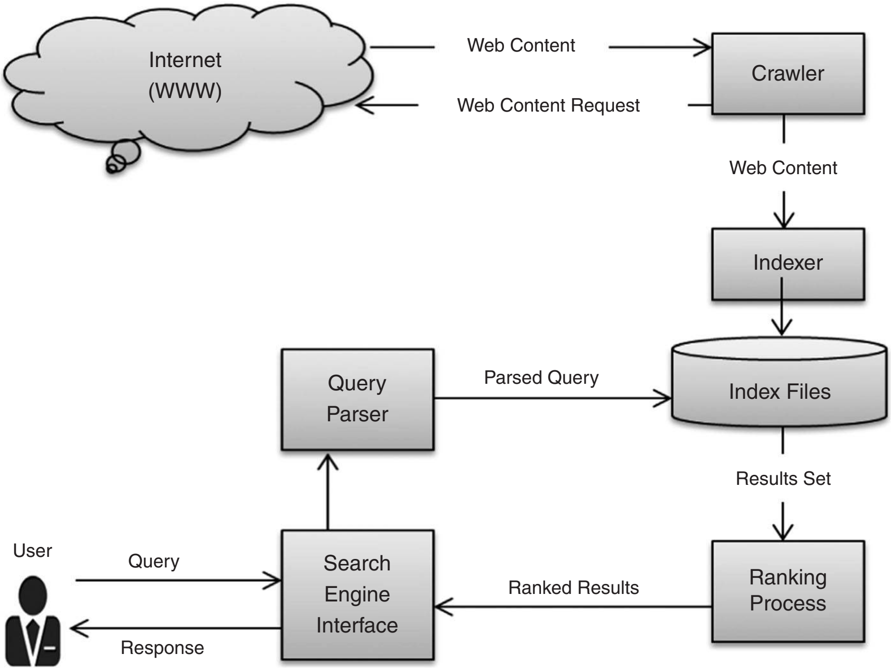

Figure 3.
Simple architecture of a typical web search engine

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

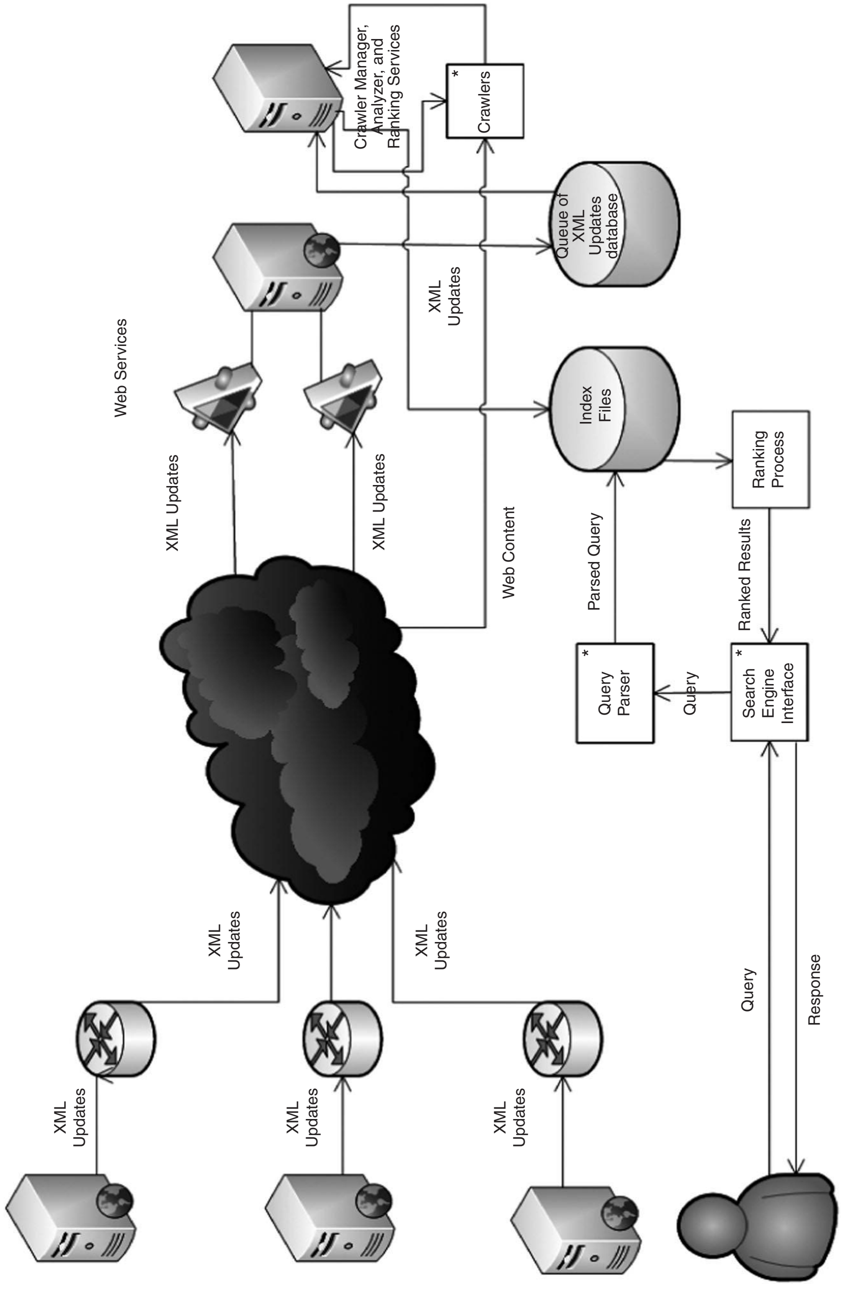

Figure 4.
Simple architecture of the proposed web search engine

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

7.2 Push method

The web server automatically monitors changes including new content on each domain and creates an XML file[1], which is different than the Sitemap's XML file. It includes the updates and the instructions (XML tags) needed by the search engine to implement the changes. However, a web file will be crawled normally as a new file, if the size of the updates required is much greater than the size of the file itself. The XML file would be pushed by the web server through web services or other means. Search engines then store, queue, and process the XML files in order to implement the changes similar to any operating system or application updates and subsequently index the refreshed content. In summary, this push method provides the search engine with the necessary instructions and the needed content (updates) in order to refresh its indexes.

To deal with different types of content (i.e. binary and text), multiple techniques could be adopted. Binary files may be processed separately from text, or text files might be converted to binary and processed as such. However, due to issues related to binary files both types should be processed separately. This research focusses on the push method.

7.2.1 Text content. Comparison between old and new text content is straightforward. Many algorithms have been developed to accomplish this task. One of the best known algorithms has been developed by Neil Fraser. "The Diff Match and Patch" is based on robust algorithms to perform the operations required for synchronizing plain text (Google Inc., n.d.). This algorithm has been used in this research.

7.2.2 Binary content. Binary files however, are more complicated than text content to synchronize. In this paper, JojoDiff (JDIFF)-a program based on a heuristic algorithm with constant space and linear time complexity, has been used for comparing binary files and generating the differences. JojoPatch (JPTCH) has been used to construct the updated files using the original file and the file containing the changes. The differences file should not be compressed when speed is a priority over accuracy (Heirbaut, 2011). JPTCH is adopted in this approach.

7.2.3 Algorithms. The pseudo-code of the change algorithm server side (will be implemented as a time-based task):

```
Begin
  Tc = number of days or time period to test files for changes;
  T0 = Application variable (StartDate);
  NumberOfRuns = application variable (n);
    If (TimeNow > = T0+n*Tc)
      For each Domain on the server
        For each file of the selected domain
          Load changed file and pre-changed file;
          Calculate the difference of both files; (Algorithm 1)
          Formulate changes as XML records;
          If the size of the difference of XML records needed is
            more than the original file or the file is eliminated
            Append a record to the domain XML file that the entire file
            needs to be downloaded or no need to update the eliminated file
          Else
            Append changes records to the domain XML file;
        End
    End For
```

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

```
    End for
    $n = n+1$;
    Save n as and application variable again
End If
End
```

The pseudo-code of the change algorithm search engine side (will be implemented as a time-based task)[2]:

```
Begin
For each server
    For each domain
    Receive the domain XML file
    Check the integrity and the authenticity of the XML file;
    If (XML authentic)
      For each file
        If file needs to be changed
        Implement changes; (Algorithm 2)
        Test pages;
        If (not pass)
          Revert to old version of file;
          Alert server;
        End If
        Else if file needs to be crawled (Size of XML records more than new file)
          Queue file for crawling
        End if
    End If
  End For
End for
End
```

The structure of the XML file suggested that its size can be calculated using Equation (1):

$$S = u + \alpha \times c + c_0$$

where u is the size of the updates, α is the number of updates or locations that have been updated, c is the needed XML code to formulate one change (108 bytes), and c_0 the default size of the empty XML file (50 bytes).

8. Experiment

The simulation was set up on a Dell PowerEdge server running Microsoft Windows server 2008 with Microsoft Internet Information Services version 7.0. A crawler was developed using ASP.Net (VB). The crawler was configured to run once every day for a sample of fixed pages from a determined sample of selected websites. A randomly selected set of websites with high traffic from domaintyper.com was used. Due to the presence of a proxy in the UAE, inaccessible websites were removed from the list. In total, 107 domains/websites from the refined set were used in this experiment. From each domain, ten pages were randomly selected to be crawled. The total pages crawled was 1,070. The same pages from the same websites were crawled every day for 22 days starting from the November 30 to the December 24, 2013. However, some days were skipped due to internet connection issues. To avoid the clutter, 55 pages were randomly

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

selected and used to present the results. Pages were numbered from one to 55. Data collected were divided into two types, binary and text based. A program was built using VB.Net, based on Neil Fraser's algorithm "Diff Match and Patch" to determine the differences between text files. The binary data were processed using an updated version of JDIFF and JPTCH. Both programs have been integrated in a program that can batch process a set of related text and binary files. UTF-8 encoding was adopted to support different languages in the crawling process and the creation of the XML file needed by the search engines and browser to update cashed/or stored files. MATLAB R2011b was used to implement the algorithms (1 and 2) used in this research; however, for clarity, processed data were exported to Microsoft Excel 2010 to generate the necessary graphs and figures when possible.

9. Results and analysis

Following the experiment steps, content crawled is divided to two parts: text and binary.

9.1 Text content

In this part, text content was processed and the results are presented. Figure 5 shows that the size of updates varies from one web page to another. Further investigation reveals that the pages that are drastically changed are from domains that are being changed often and with considerable content such as CNN.com, Amazon.com, Yahoo.com, and ebay.com. However, other web pages, such as IBM.com, Microsoft.com, and other similar sites are changing less often and with smaller amounts of content. The domains that have been changing at a speedy rate and with considerable content

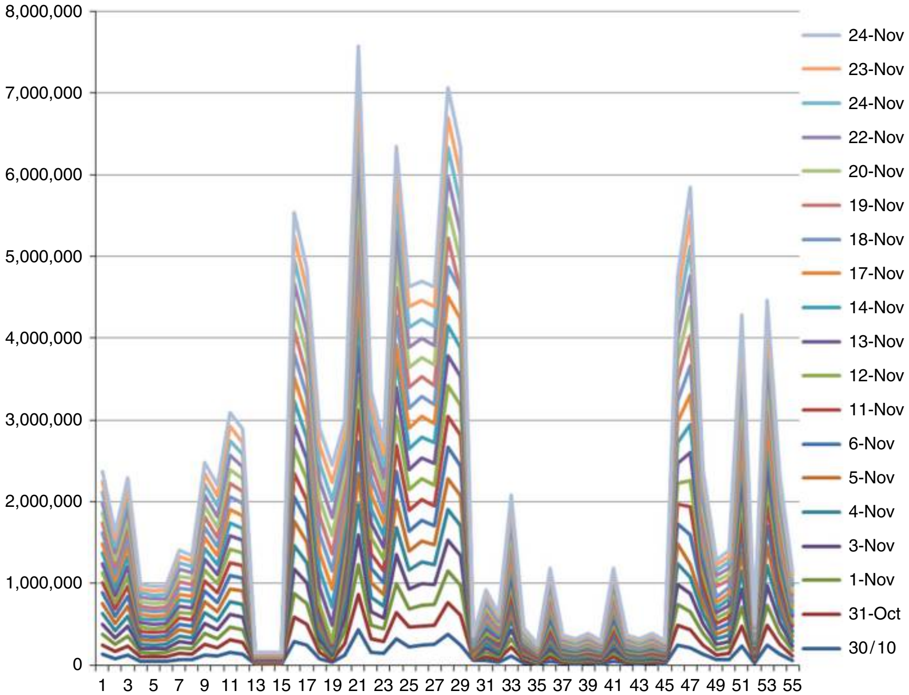

Figure 5. Updates in bytes for each page for 19 days

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

should be crawled often by search engines or the server should push such updates on a daily basis using the proposed approach in this research.

Figure 6 shows clearly that the proposed approach offers considerable gain when it comes to the data that need to be transmitted or exchanged between web servers and one search engine.

Figure 7 shows the percentage gained for each page. Let s_i be the size of new pages plotted in Figure 7 that needs to be crawled by search engines without the use of the proposed approach (current crawling method used by search engines), u_i be the size of updates plotted in the same figure required by the search engine to index the same pages using the proposed approach without compression of the XML file, and n the number of pages. The average traffic reduced can be calculated using the following equation:

$$\langle S \rangle = \frac{\sum_i^n u_i}{\sum_i^n s_i} \times 100 \tag{2}$$

Subtracting the result produced by Equation (2) from 100 percent suggests that the average gain is approximately 25 percent. However, the gain for some pages reaches more than 90 percent.

Figure 8 depicts the overhead in size necessary to formulate the XML file needed to provide instructions to search engines or browsers to update cached or stored content. Let x_i be the size of the XML files plotted in the Figure 8, u_i be the update size plotted in the same figure, and n the number of pages. The average size of the overhead O introduced by the use of XML can be calculated using the following equation:

$$\langle O \rangle = \frac{\sum_i^n (x_i - u_i)}{\sum_i^n x_i} \times 100 \tag{3}$$

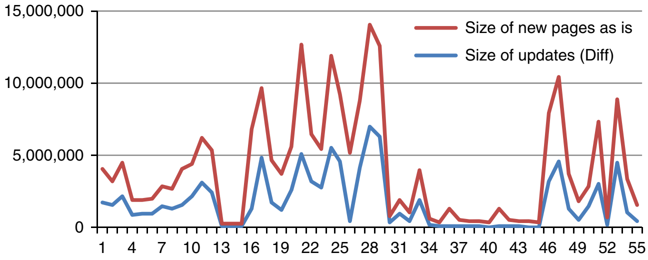

Figure 6. Comparison between updates and the modified page in bytes for 19 days

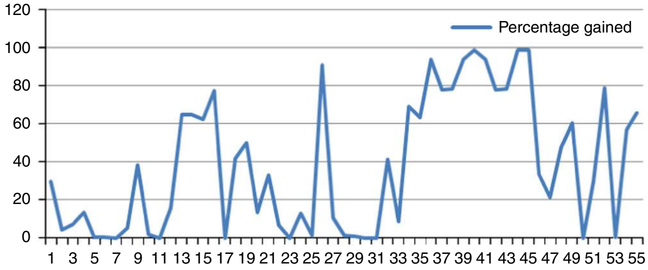

Figure 7. Percent of traffic gained for each page for 19 days

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

Subtracting the result produced by Equation (3) from 100 percent suggests that the overhead is of the order of 5 percent on average, which is considerably lower than the gain.

To further improve the effectiveness of the proposed approach, the XML file might be compressed. The size of the XML file generated on the web server approach varies from one domain to another. Usually it is more than 20 kb, and there is no need for data querying[3] because the file for each domain is processed once. For faster compression and moderate compression ratio, XMill combined with one of the back-end general purpose compressors, such as gzip, bzip2, or prediction by partial matching might be considered as the best choice (Sakr, 2009). The compression ratio (bits/bytes) changes according to the size and the type of the file to be compressed (Liefke and Suciu, 2000). Assuming the XML file for each domain is more than 1,000 kb, the compression might reach up to 50 percent when XMill is used (Liefke and Suciu, 2000). In our case, because of the structure and the high level of redundancies and other characteristics of the XML files that need to be pushed to search engines or browsers, XMill algorithm did reduce the size of the XML files up to 93 percent (Figure 9). To compute the average size

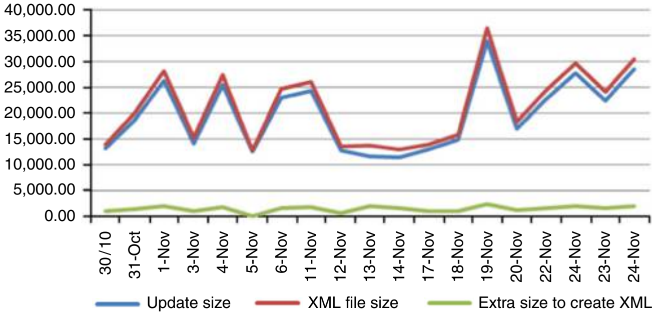

Figure 8.
Comparison between initial updates and the XML file that needs to be exchanged with the search engine or the browser in bytes for each page for 19 days

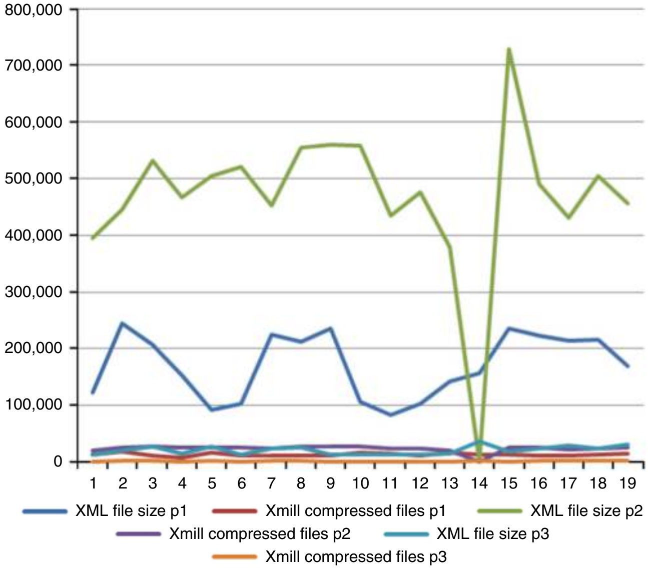

Figure 9.
Comparison between the initial XML file and compressed XML file using XMill algorithm

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

reduction, let x_i be the size of the XML files plotted in the Figure 8, c_i be the compressed version of the same file, and n the number of pages. The average size R reduced using compression can be computed using the following equation:

$$\langle R \rangle = \frac{\sum_i^n c_i}{\sum_i^n x_i} \times 100 \tag{4}$$

The approach in this study without compression reduces the updates on average by 25 percent (Equation (2)), which means only 75 percent of the data is required to index updated pages. XMill algorithm, when adopted, reduced the XML file generated on average by 80 percent (Equation (4)). Consequently, the changes or updates of text content can be reduced up to 80 of 75 percent which equal 60 percent (Equation (5)). Therefore, on average only 15 percent (75-60 percent) of the updates are required for search engines to update their existing text-based content. In addition, document type definition and other extra communication required in the indexing process are estimated to be less than 1 percent. Therefore on average only 16 percent is required to refresh text web content indexed by search engines:

$$\langle C \rangle = \left(\frac{80}{100} \times \frac{75}{100}\right) = 60\% \tag{5}$$

9.2 Binary content

The same algorithms (1, 2) were applied to binary files. However, the comparison algorithm is different. In this paper, JDIFF and JPTCH were used. A random sample of pictures from different domains was tracked for changes (161 binary files).

The size of changes generated by JDIFF algorithm is 0 when no changes are done to the binary file (Figure 10). If the image is replaced, the new file size value is 0, which suggests the new image should be uploaded. The size of binary files might increase or decrease, therefore, the absolute function is used to keep all values positive. This study is concerned more about the size than its sign. As mentioned by the proposed algorithm in this research, if the file size generated by JDIFF is greater than or equal to the size of the new binary file, the new binary file should be indexed as usual in order to minimize the traffic and the processing time.

Figure 11 shows that most of the files are the same or removed. However, when binary files changed, the size of the file needed to recreate the new file using JPTCH is almost equal to the size of the new file that needs to be updated between servers and search engines or browsers.

The same pattern was observed in figures concerning the days 3-4, 5-6, 7-8, 9-10, 11-12. Most of the images were the same or replaced by new ones. However, for images that have been changed, the traffic saved is not worth the processing time, at least for the JDIFF and JPTCH algorithms.

10. Discussion

New internet services have been emerging and evolving exponentially, such as video conferencing; IP telephony; HD video on demand; hologram-based application; radio broadcasting and music streaming. Technology and content convergence has been promoting the creation of a substantial amount of content. Webcam, cameras, smart phones, etc. have been the essence of digital content creation. A major portion of the

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

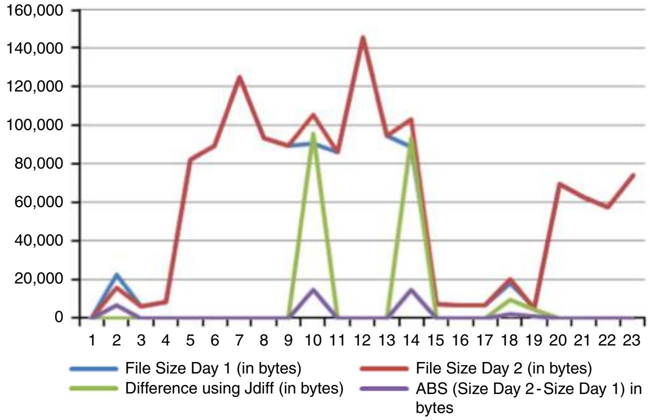

Figure 10.
Comparison between initial binary file and new binary file using JojoDiff algorithm for day 1 and day 2

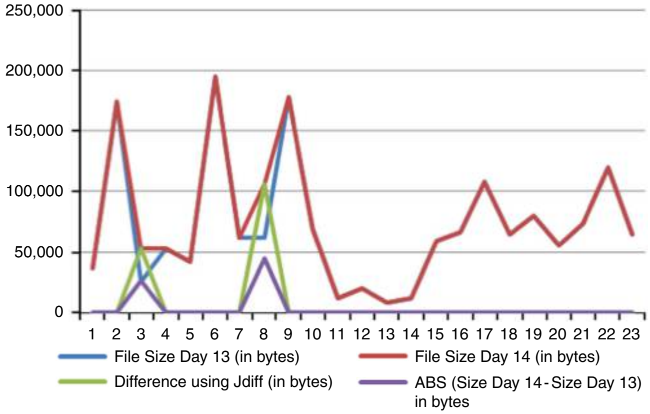

Figure 11.
Comparison between initial binary file and new binary file using JojoDiff algorithm for day 13 and day 14

amount of data created eventually has to be transmitted. However, the internet, the underlying infrastructure and the services available, will eventually reach their fundamental caps. Solutions to overcome today's internet limitations have been emerging. New versions of superfast networks based on new and optimized protocols have been developed, such as Internet Project[2] in the USA; Australian academic and research network; British academic network (JANET); the pan-European data network for the research and education community (GÉANT); French research network (Renater); Dutch research network (SURFnet), and others. However, access to these networks has been limited to researchers and educational institutions. In parallel with these efforts, content has been exponentially generated using the traditional internet network and its services. It has to be effectively managed to ease the stress on the internet infrastructure. Search engines have to crawl, collect, store, process, and index this content. This process has been generating high networking traffic between host servers and the search engines' networks. The estimated web text content crawled by Google spiders is massive (8.44 pb/day). The daily network traffic generated by Google's crawlers and other similar search engines has to be optimized.

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

Innovative techniques have to be developed and implemented within the existing and new networks. The proposed approach in this paper is in line with this direction. Our experiment showed that the proposed approach decreased the text traffic generated by search engines by approximately 84 percent. That is 7.09 pb/day of traffic created by Google's crawlers will be reduced. Further enhancement to existing algorithms can help cut the traffic even more. However, binary files are still causing issues. The processing time of binary files and the error rate might overshadow the benefits when it comes to decreasing traffic. New binary comparison algorithms have to be developed in order to optimize the traffic caused by search engines when crawling sites with many binary files.

Despite issues related to binary files, the proposed approach may solve many problems. The automation of the process will mitigate the issues related to manual handling and submission of Sitemap using Google's webmasters tools. The frequency of crawling might be improved, due to the shift of the control from search engines to the web servers. Web servers with fewer updates, presumably, will submit updates less often and subsequently, blind crawling will be avoided. Processing power needed to implement this approach will be shared between search engines and internet service providers' infrastructures. Web servers from all over the internet have the opportunity to submit their updates and therefore, deep web content will be available for search engines and therefore for users.

Certainly, the proposed approach would significantly reduce the traffic caused by crawlers and mitigate the deep web issue. Its implementation and adoption by web browsers would help eliminate staggering internet traffic by enabling browsers to refresh their cache using the same mechanisms discussed herein. Further experimental research, however, is required to support this claim.

11. Conclusion

It is clear, therefore, that crawlers consume a considerable part of the internet network bandwidth. Search engines must ensure that effective crawling of web content is achieved. The web content explosion caused by Web 2.0 and the increased number of internet users using different types of content have been challenging the available bandwidth. Excess traffic by crawlers has to be eliminated. Furthermore, search engines are indexing a small portion of available web content. A considerable part called deep web has not been indexed by search engines, thus, invisible and inaccessible by search engines' users. New approaches are required to mitigate the traffic issue caused by search engines' crawlers and deep web predicament. The experimental implementation of the approach proposed in this study demonstrated that the traffic caused by crawlers can be decreased by up to 84 percent for text content. Moreover, the deep web issue would be mitigated. However, binary files require the development of dexterous comparison and difference calculation algorithms to be successful. Above all, in order to alleviate the traffic problem and to maintain an acceptable level of available internet bandwidth for online users, search engines have to find new ways to crawl the web. Cooperation between web servers and search engines as proposed in this study is a must. The invisible part of the web has to be crawled and indexed; if not, the internet and what it stands for is in jeopardy. The proposed approach in this study has focussed on crawlers. Yet, web browsers could benefit from the presented approach and staggering internet traffic will be reduced. However, experimental research is required to confirm the later claim.

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

Notes

1. This is a special XML file. It is structured according to the algorithm proposed in this study.
2. The same algorithm might be adapted for browsers requests.
3. The XML file is parsed and processed as whole. No specific data queries are required by the search engine. If the file is queried frequently for data, compression will be costly performance-wise.

References

Anagnostopoulos, L. and Stavropoulos, P. (2011), "On the feasibility of applying capture – recapture experiments for web evolution estimations", *Applied Mathematics Letters*, Vol. 24 No. 6, pp. 1031-1036.

Argaez, E. (n.d.), "Finding information in the internet", available at: www.internetworldstats.com/articles/art028.htm (accessed February 2013).

Badawi, M., Mohamed, A., Hussein, A. and Gheith, M. (2013), "Maintaining the search engine freshness using mobile agent", *Egyptian Informatics Journal*, Vol. 14 No. 1, pp. 27-36.

Bergman, M.K. (2001), "White paper: the deep web: surfacing hidden value", *Journal of Electronic Publishing (JEP)*, Vol. 7 No. 1.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A. and Wiener, J. (2000), "Graph structure in the web: experiments and models", *Proceedings of the 9th international World Wide Web conference on Computer networks, The International Journal of Computer and Telecommunications Networking*, May 15-19, Amsterdam, pp. 309-320.

Brodkin, J. (2012), "Bandwidth explosion: as internet use soars, can bottlenecks be averted?", available at: www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/VNI_Hyperconnectivity_WP.html (accessed January 2014).

Cho, J. and Molina, H.G. (2000), "Estimating frequency of change", available at: http://ilpubs.stanford.edu:8090/471/1/2000-4.pdf (accessed May 2012).

Cho, J. and Molina, H.G. (2012), "The evolution of the web and implications for an incremental crawler", *Proceedings of the 26th International Conference on Very Large Databases, 2000*, available at: http://oak.cs.ucla.edu/~cho/papers/cho-evol.pdf (accessed May 2013).

Cisco (2013), "Cisco visual networking index: forecast and methodology, 2012-2017", available at: www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf (accessed January 2014).

Du, Y., Pen, Q. and Gao, Z. (2013), "A topic-specific crawling strategy based on semantics similarity", *Data & Knowledge Engineering*, Vol. 88, November, pp. 75-93.

Eijk, N.V. (2010), "Search engines, the new bottleneck for content access", in Preissl, B., Haucap, J. and Curwen, P. (Eds), *Telecommunication Markets, Drivers and Impediments*, Springer, pp. 141-157, available at: http://ssrn.com/abstract=1609850 (accessed May 17, 2010).

Evans, P.M. (2007), "Analysing Google rankings through search engine optimization data", *Internet Research*, Vol. 17 No. 1, pp. 21-37.

Ferreira, R., Freitasa, F., Britob, P., Meloa, J., Limaa, R. and Costab, E. (2013), "RetriBlog: an architecture-centered framework for developing blog crawlers", *Expert Systems with Applications*, Vol. 40 No. 4, pp. 1177-1195.

Garg, P. (2008), "One standard fits all: robots exclusion protocol for Yahoo!, Google and Microsoft", available at: www.ysearchblog.com/2008/06/03/one-standard-fits-all-robots-exclusion-protocol-for-yahoo-google-and-microsoft/ (accessed October 2013).

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

Google Inc. (n.d.), "Google-diff-match-patch",available at: https://code.google.com/p/google-diff-match-patch/ (accessed May 2013).

Gopinah, S. (2005), "Structural and conceptual changes in the information landscape: the new challenges for information professionals", available at: dspace.iimk.ac.in/bitstream/2259/251/1/07-saji-paper.pdf (accessed March 2012).

Heirbaut, J. (2011), "JojoDiff – diff utility for binary files", available at: http://jojodiff.sourceforge.net/ (accessed May 2013).

Internet World Stats (IWS) (2015), "Internet usage statistics", available at: www.internetworldstats.com/stats.htm (accessed December 2015).

Jiang, J., Song, X., Yu, N. and Lin, C. (2013), "FoCUS: learning to crawl web forums knowledge and data engineering", *IEEE Transactions*, Vol. 25 No. 6, pp. 255-268.

Ke, Y., Deng, L., Ng, W. and Lee, D. (2006), "Web dynamics and their ramifications for the development of web search engines", *Computer Networks*, Vol. 50 No. 10, pp. 1430-1447.

Koster, M. (1994), "A standard for robot exclusion", available at: www.robotstxt.org/orig.html (accessed October 2013).

Laudon, C.L. and Traver, G.C. (2008), *E-Commerce: Business, Technology, Society*, 4/E, Prentice Hall.

Lawrence, S. and Giles, L. (1999), "Accessibility of information on the web", *Nature*, Vol. 400 No. 6740, pp. 107-109.

Li, Y., Wang, Y. and Du, J. (2013), "E-FFC: an enhanced form-focused crawler for domain-specific deep web databases", *Journal of Intelligent Information Systems*, Vol. 40 No. 1, pp. 159-184.

Liefke, H. and Suciu, D. (2000), "XMill: an efficient compressor for XML data", *Proceedings of the 2000 ACM SIGMOD International Conference on Management of data, ACM, New York, NY*, pp. 153-164.

Liu, H. and Milios, E. (2012), "Probabilistic models for focused web crawling", *Computational Intelligence*, Vol. 28 No. 3, pp. 289-328.

Liu, W. and Du, Y. (2014), "A novel focused crawler based on cell-like membrane computing optimization algorithm", *Neurocomputing*, Vol. 123 No. 10, pp. 266-280.

Madaan, R., Dixit, A., Sharma, A.K. and Bhatia, K.K. (2010), "A framework for incremental hidden Web crawler", *In International Journal on Computer Science and Engineering*, Vol. 2 No. 3, pp. 753-758.

Misra, P.N. and Sorenson, H.W. (1975), "Parameter estimation in poisson processes", *IEEE Transactions on Information Theory*, Vol. 21 No. 1, pp. 87-90.

Myllymaki, J. (2002), "Effective web data extraction with standard XML technologies", *Computer Networks*, Vol. 39 No. 5, pp. 635-644.

Niwase, H., Araki, H., Takada, N., Nakayama, H., Sugiyama, A., Kakue, T., Shimobaba, T. and Ito, T. (2013), "Time-division electroholography of the three-dimensional object", *Proceedings of Three Dimensional Systems and Applications: 3DSA2013*, Osaka, June 26-28, pp. 4-2.

Onural, L., Sikora, T., Ostermann, J., Smolic, A., Civanlar, R. and Watson, J. (2006), "An assessment of 3DTV technologies", *Proceeding of NAB 2006*, pp. 456-467.

Peisu, X., Ke, T. and Qinzhen, H. (2008), "A framework of deep Web crawler", *Proceedings of the 27th Chinese Control Conference*, *Kunming and Yunnan*.

Perez-Sola, C. and Herrera-Joancomarti, J. (2013), "OSN crawling schedulers and their implications on k-plexes detection", *International Journal of Intelligent Systems*, Vol. 28 No. 6, pp. 583-605.

Press release (2014), "comScore releases December 2013 US search engine rankings", available at: www.comscore.com/Insights/Press_Releases/2014/1/comScore_Releases_December_2013_US_Search_Engine_Rankings (accessed January).

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)

Punam, B., Anjali, T. and Hema, B. (2013), "Focused crawling of tagged web resources using ontology", *Computers & Electrical Engineering*, Vol. 39 No. 2, pp. 613-628.

Risvik, M.K. and Michelsen, R. (2002), "Search engines and Web dynamics", *Computer Networks*, Vol. 39 No. 23, pp. 289-302.

Sakr, S. (2009), "XML compression techniques: a survey and comparison", *Journal of Computer and System Sciences*, Vol. 75 No. 5, pp. 303-322.

Sharma, A.K., Gupta, J.P. and Agarwal, D.P. (2003a), "A novel approach towards management of volatile information", *Journal of CSI*, Vol. 33 No. 1, pp. 18-27.

Sharma, A.K., Gupta, J.P. and Agarwal, D.P. (2003b), "Augment hypertext documents suitable for parallel crawlers", *Proceeding of a National workshop on Information Technology Services and Applications (WITSA), February 27-28, New Delhi*.

Sharma, A.K., Gupta, J.P. and Agarwal, D.P. (2010), "PARCAHYD: an architecture of parallel crawler based on augmented hypertext documents", *International Journal of Advancements in Technology*, Vol. 1 No. 2, pp. 207-283.

Singhal, N., Dixit, A. and Sharma, K.A. (2010), "Design of a priority based frequency regulated incremental crawler", *International Journal of Computer Applications*, Vol. 1 No. 1, pp. 42-47.

Spink, A., Bernard, J., Jansen, J.B., Kathuria, V. and Koshman, S. (2006), "Overlap among major web search engines", *Internet Research*, Vol. 16 No. 4, pp. 419-426.

Sullivan, D. (2012), "Google: 100 billion searches per month, search to integrate Gmail, launching enhanced search app for iOS", Search engine land, August 8.

Taylor, M.H. and Karlin, S. (1998), *An Introduction To Stochastic Modeling*, 3rd ed., Academic Press, San Diego, CA.

Torkestani, A.J. (2012), "An adaptive focused Web crawling algorithm based on learning automata", *Applied Intelligence*, Vol. 37 No. 4, pp. 586-601.

Uemura, Y., Itokawa, T., Kitasuka, T. and Aritsugi, M. (2012), "An effectively focused crawling system", *Studies in Computational Intelligence*, Vol. 376, pp. 61-76.

Winkler, L.R. (1972), *An Introduction to Bayesian Inference and Decision*, 2nd ed., Holt, Rinehart and Winston, Inc., Austin, TX.

Yalçin, N. and Köse, U. (2010), "What is search engine optimization: SEO?", *Procedia Social and Behavioral Sciences*, Vol. 9 No. 2010, pp. 487-493.

Yan, H., Wang, J., Li, X. and Guo, L. (2002), "Architectural design and evaluation of an efficient Web-crawling system", *The Journal of Systems and Software*, Vol. 60, pp. 185-193.

Yuan, X. and Harms, J. (2002), "An efficient scheme to remove crawler traffic from the internet", *Proceedings of the 11th International Conferences on Computer Communications and Networks, October 14-16 Miami, Florida*, pp. 90-95.

Zeifman, I. (2013), "Bot traffic is up to 61.5% of all website traffic", available at: www.incapsula.com/blog/bot-traffic-report-2013.html (accessed October 2014).

Zheng, Q., Wu, Z., Cheng, X., Jiang, L. and Liu, J. (2013), "Learning to crawl deep web", *Information Systems*, Vol. 38 No. 6, pp. 801-819.

Further reading

Prakash, J. and Kumar, R. (2015), "Web crawling through shark-search using pagerank", *Procedia Computer Science*, Vol. 48, pp. 210-216.

Nath, R., Bal, S. and Singh, M. (2007), "Load reducing techniques on the websites and other resources: a comparative study and future research directions", *Computer Journal of Advanced Research in Computer Engineering*, Vol. 1 No. 1, pp. 39-49.

Singhal, N., Agarwal, R.P., Dixit, A. and Sharma, A.K. (2011), "Information retrieval from the web and application of migrating crawler", *Proceedings of international conference on computational intelligence and communication systems*, October 7-9, Gwalior, pp. 480-483.

Sitemaps.org (2008), "What are Sitemaps?", available at: www.sitemaps.org/ (accessed October 2013).

About the author

Dr Mhamed Zineddine was born and raised in Morocco. After graduation from the Mohammed V University Rabat, he worked as a Programmer for three years. He spent ten years in the USA. He got his Master's Degree from the IONA College New Rochelle, NY and his PhD from the Capella University, MN, USA. He worked as a Web Developer, a Network Designer, and a System Engineer for years before he joined the ALHOSN University in 2008 as an Assistant Professor. He was assigned many administrative positions including the Chair of the Management Information Systems Department and the IT Director. He received the 3rd Asia's Best Business Schools award as the Best IT Professor in 2012. Dr Zineddine currently lives in Abu Dhabi, UAE. He is interested in IT security, cyberwar, optimization, and healthcare information security and privacy. Dr Mhamed Zineddine can be contacted at: z5868@yahoo.com

For instructions on how to order reprints of this article, please visit our website:
www.emeraldgrouppublishing.com/licensing/reprints.htm
Or contact us for further details: **permissions@emeraldinsight.com**

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 20:29 09 November 2016 (PT)