On Benefits Of Using A Web Browser As A Distributed Crawler
Sergey Fedorenko

ABSTRACT
The current use of crawlers for information retrieval from the web is reviewed and found as inadequate and inherently flawed. The new model for information retrieval from the web is discussed where instead of crawlers millions of web browsers that are used to surf the web are utilized. The simulation of the model is used to estimate some of its possible benefits.

Keywords
Search engine, crawler, browser, web.

1. INTRODUCTION
The World Wide Web provides enormous wealth of information and capabilities for collaboration between its users. These capabilities are unparalleled by anything else – it is the global wide communication network and the database that is distributed between of millions of computers. Increasingly, it is playing an important role in the global economy and is being used for a growing number of applications. One of the most fundamental usages of the web and the one that many other usages are based upon is information retrieval. The task of finding relevant information quickly is of paramount importance and is generating large amount of investment and research. Currently, the common way to find information on the web is by using search engines.

A typical search engine tries to index pages on the web in a database of the search engine. Using information stored in this database the search engine processes the search queries and generates responses to the query. A typical search engine consists of three major conceptual parts: a web crawler, an indexer, and a query processor [1]. Web crawlers (also called robots or spiders) perform information harvesting by traversing the web and retrieving pages from the web. The efficiency of a crawler is of utmost importance since it determines the usability of the search engine that employs the crawler – to be useful the crawler needs to harvest a page in a fraction of a second since the number of pages on the web is measured in billions and amount of information that the web contains is measured in terabytes [1,2]. To make things worse the information on the web is being updated frequently, thus crawlers need to traverse the same area of the web repeatedly in order to avoid dead links and outdated information [1,2]. When the pages that have been harvested by crawlers are stored by a searched engine for indexing, then other non-trivial challenges face implementers of the search engine such as storing enormous amount of information that is measured in terabytes and creating gigantic index trees that cannot be stored in the physical memory on a single computer due to the sizes of the indices [1,3].

In the current state of affairs, a typical search engine is distributed across array of hundreds of computers that provide processor power to crawl the web, digesting the pages that were harvested, generating indices, and storage for the pages. However, parallelization of the web page harvesting process is not enough to cover the entire web. It is estimated that the best search engines cover about 10% of the part of the web that consists of static HTML pages. The efficiency of crawlers is constantly being improved, processing power of the computers is growing as is the storage capabilities, but that does not guarantee that the crawlers will eventually catch up because the web is growing exponentially and there is no indication that this growth is slowing down [1].

More importantly, there is a fundamental limitation on what crawlers can retrieve from the web based on the structure of the web. The web has been found to consist of the following parts: strongly connected central core (27%), the component OUT consisting of pages that can be reached from the core but do not connect back to the core (21%), component IN that consists of pages that can reach the core but not from it (22%), the TENDRILS that cannot reach the core and cannot be reached from it (22%), and other disconnected components (8%) [4]. This structure of the web effectively prevents crawlers from harvesting pages from IN, TENDRILS, and other disconnected components making more than 50% of the web inaccessible for the search engines [5].

Another fundamental problem is that crawlers cannot fill in web forms and submit requests, which is necessary in order to access many dynamically generated pages. It is estimated that dynamically generated HTML pages outnumber static ones 100 to 1 [6]. Thus, at best a search engine of today can cover only a small portion of the web. This problem is only going to get worse as more and more web sites are starting to use dynamically generated pages to access data in databases [5].

In addition, it is estimated that crawlers generate up to 40% of the traffic on the Internet [5]. Therefore, increase in crawling the web to increase its coverage will result in generating even more traffic that chokes useful communications on the Internet. Knowing that crawlers take away significant portion of the bandwidth from legitimate users many web sites prohibit crawlers from accessing their pages, which makes yet another portion of the web inaccessible to search engines.

The researchers in this field are realizing the depth of the problem and are coming with new approaches to improve information retrieval from the web. The significant amount of research has been performed to improve crawling strategies and optimize information retrieval [7]. Recent research has been generating such ideas as employing crawlers that are running locally on web servers [8,10], collaboration between web servers and search engines where data is ‘pushed’ from web servers to a search engine [11], interception of the information as it passes through the networks [5], and employment and refinement of metasearch where search query is processed by a number of search engines [9]. However, there appears to be no new approach that have generated significant amount of positive feedback as a definitive solution for the problem. Nevertheless, it is clear that significant amount of research and thought is gravitating around an idea that
the improvement of the information retrieval from the web has to
do with making search engines more distributed in nature.

In addition, it must be mentioned that during the last few years
there was a significant number of distributed web applications
produced by the industry that explored possibilities of information
retrieval and exchange on the web. One type of distributed web
applications is peer-to-peer file sharing applications such as
Napster, Gnutella, KaZaA [http://www.kazaa.com], Morpheus
http://www.musicity.com, Myster [http://www.homepage.mac.com/myster/], and many other variants
that use similar concept. These applications provide to a different
extent distributed search capabilities. However, these search
capabilities are limited to the applications’ specific peer-to-peer
subscriber networks. Another type of web applications is
distributed searching environments such as Human-Links
http://www.human-links.com and NeuroGrid
http://www.neurogrid.net where data is organized in associative
manner instead of hierarchical and the application functions as a
shared neural network. Again, these applications are limited to
peer-to-peer networks that are set up at company or a community
level as opposed to encompassing the entire web or its significant
segments. Finally, there are applications such as Grub
http://grub.org that are utilizing distributed crawlers that attempt
to index the entire web by enlisting volunteers that will install a
client that crawls the web when host machine where it is installed
is idle. The client crawler runs as a screen saver for example to
cycle unused CPU cycles and bandwidth by crawling a portion
of the web and harvesting pages to a centralized database where
index is kept. This is the same configuration as with traditional
search engines with the only significant difference that computers
that host crawlers are ‘donated’ and are distributed across
numerous locations instead of being ‘farmed’ at a single or few
locations.

After reviewing the state of affairs in the industry and of the
current research in the field it is evident that currently there is no
solution to retrieve information from the most of web fast and
efficiently. At best the information is retrieved from a small
segment of the web and this limitation is a result of deficiencies of
the current model for information retrieval from the web.

The purpose of this paper is to propose and discuss a different
model for information retrieval that potentially can improve the
coverage of the web, provide better scalability to accommodate
for growth of the web, and decrease expense of the process of
information retrieval from the web. Since the web is distributed,
is so immense, and is growing so fast it is likely that a successful
solution to improve retrieval of information from the web ought to
mirror these properties of the web – it should also be distributed
on the scale that is comparable with the scale of the web itself and
provide scalability to accommodate for the growth of the web.

2. UTILIZING WEB BROWSERS OF THE
INTERNET USERS AS A DISTRIBUTED
CRAWLER

At the heart of the proposed solution is an idea to create the
ultimate search engine that is distributed among and harnesses
power of millions of computers that are being used to surf the web
by using a web browser for functions performed by crawlers.
Since every browser has to ‘harvest’ a page before showing it to a
user, thus, a browser already performs a function of a crawler –

let’s just utilize it. A web browser can also be used to digest the
information of the ‘surfed’ pages for later indexing at a
centralized database where browsers send the index information
extracted from pages that were ‘surfed’.

The proposed approach unloads search engine’s crawling task to
the millions of client machines that continuously scour the web,
allows using processing power of these remote machines to
extract the information from a web site that is being currently
visited by a web browser. Since the extraction of information from
visited pages is occurring in the web browser, there is no need to
store these pages on the central location computers of the search
engine. Thus, the proposed approach may significantly alleviate
three difficult problems of retrieval of information from the web –
insufficient efficiency to harvest information from the web by
crawlers, enormous requirements for storage of harvested pages,
and requirements for processing power to extract the information
from the pages. The coverage of the web should improve with the
introduction of this approach since users can access parts of the
web that crawlers cannot and it is reasonable to make conjecture
that vast majority of the web pages has been accessed at least once
by using a web browser – at least creator of the pages must have
tested web site and looked at its pages. Also new problems are
introduced. There is a need to communicate extracted information
from web browsers-crawlers to a central location where index is
maintained and where search queries are processed. There are
issues with privacy of users whose browsers are used to harvest
information from the web. These issues are discussed in the
following section.

2.1 Model

The new model for the process of information retrieval from the
web has the process consisting of the three major conceptual
stages: information harvesting by a web browser at a client’s
location, delivery of the information from a web browser to search
engine’s central database, and preprocessing, storage and retrieval
of the information at the central location. Each of these stages
allows a different degree of variation of its concept.

The information harvesting that is performed by a web browser of
a user surfing the web can be page or site oriented. When it is
page oriented a browser-crawler is only harvesting pages that
were downloaded by a user explicitly. The only thing that is
different from conventional browser is that the downloaded pages
are sent to the central location. This mode is using a least amount
of resources on the client machine. When a browser-crawler is site
oriented then it crawls in a background the site that the user
pointed a browser to. The crawling is performed while the user is
viewing already downloaded page, it stops when the user points to
a different page, and resumes when the page requested by a user is
downloaded and can be viewed by the user. Hence, the crawling is
transparent to a user and the only difference besides sending of
the downloaded pages to the central location is that the client’s
CPU cycles are utilized that would be wasted with conventional
browser. When the site is completely crawled then a browser-
crawler continues breadth-first search (BFS) crawling of sites
connected to the one that was completely harvested until a user
points the browser to a new web site. The immediate benefit to the
user would be a cache of the crawled pages. Since it is likely that
a user will want to view more than one page of a visited web site
then his/her surfing experience will be enhanced by loading pages
faster from the cache.
The information harvesting that is performed by a web browser of a user surfing the web can also involve information extraction. This means that text of the downloaded page is processed to extract words that will be part of the index of the search index at the central location. The information extraction is relatively quick and can cut down the size of the information that needs to be sent to the central location by up to half since white space, punctuation, articles and some other parts of text are not indexed.

The delivery of the information from a web browser to search engine’s central database can either be mandatory one-way communications or it could be regulated delivery using two-way communications. When using mandatory delivery browser-crawler sends information extracted from each page to the central location unless the page is revisited during the same session and is the same as the one in the cache. When using regulated delivery browser-crawler sends a single request per site to the central location with the URL of the site that the user points to. The central location checks in the database to determine whether the site needs to be crawled (if the site has never been indexed before or if the information has not been updated recently), and sends response to the browser-crawler. If the site needs to be crawled then the browser-crawler continues with the BFS crawl. If the site has already been indexed recently, then browser-crawler starts crawling sites that are connected to the current one (if the central location indicates that these sites need to be crawled) until the user points the browser to a different site. This scheme avoids sending duplicate information and generating associated unnecessary traffic that would be resultant of situation when many users visit the same popular sites.

Another opportunity for optimization may be capability of the central location to direct browser-crawlers to crawl web sites of interest while user browses a web site that is already completely harvested by other browser-crawlers.

There are also different models for preprocessing, storage and retrieval of the information at the central location. It is possible to employ a proprietary solution similar to current search engines with significant difference that it would be using its browser-crawler that are either made by the same company or are affiliated through some agreement between search engine and browser manufacturer. Another possibility is standard communication protocol between browsers and search engines allowing discovery of nearest configured ‘central’ location through DNS or some other mechanisms. This model would allow search engines to concentrate on processing of the information and browsers to provide browsing and information harvesting services. The benefit to browser manufacturers to provide information-harvesting services could be some ‘click-based’ payment scheme similar to schemes that are currently employed by web advertisers. This way search engine manufacturer could be interested and ready to pay for using browser with large usage base that covers significant portion of the web and provides constant update of the data since that allows selling the services provided by the search engine. And the manufacturer of the browser benefits by having revenue for each crawled page or site. Finally, another option is to have web indexing facilities and the database being built intrinsically into the Internet and the web. There could be installed key servers on the Internet where each server provides an index of a selected segment of the web.

After reviewing these possible variations of the stages of the process of information retrieval from the web the new model for information retrieval can be defined as follows. Web browsers extract essential text information from pages downloaded by a user and transmit the information to a central database. While a user of browser-crawler views downloaded pages the browser-crawler performs BFS of the web site and its web neighborhood until user points to different web site. The standard protocol for communication between browser-crawlers and central information processing database systems needs to be defined to provide means for browser-crawlers to discover central database and to transmit information from pages that were harvested. The protocol allows a browser-crawler to query central database whether a site selected by a user of the browser-crawler needs to be indexed or not to avoid duplication of the information at the central database and unnecessary traffic. The protocol also provides means for a browser-crawler to inform central database when crawling of a web site has been completed to provide the central database with information that the web site has been completely harvested and does not need to be re-crawled until the web site becomes ‘stale’ after some period of time.

The new model provides another possibility to improve its effectiveness by using a combination mode for information harvesting where browser-crawlers are used as the main source of the harvested pages and traditional crawlers are employed to gather information from the parts of the web that are accessible to crawlers but which users usually don’t visit or visit infrequently. This option is especially useful with the new model of the information retrieval since it provides a dynamic picture of usage of the entire web in near real time and traditional crawlers may be directed to parts of the web that is not currently being explored by users of browser-crawlers.

2.2 Issues
One of the biggest issues that may become an obstacle for usage of browser-crawlers is the issue of privacy. For the most part this paper sidesteps privacy issue, but below are few observations about this topic. Many of web users view their web surfing information as private and are concerned if such information is accessed by somebody else. Nevertheless, all web users whether they understand it or not have to deal with situation where the privacy is rather contractual and limited in nature. Web browser store this private information on computer’s hard drive and this information can be accessed by other people unless the person to whom this information pertains secures the information. Internet service providers (ISPs) store logs that detail surfing information of the subscribers. While this information is not publicly accessible the ISP may use this information and share it with third parties such as other businesses, law enforcement agencies, and other interested parties that ISP deems as authorized to access the data. Web servers that web surfers download web pages from may log access as well and if the IP address of the web surfer is not dynamically assigned then finding name and other personal information of the person who the IP address is registered with may be a relatively easy task.

In a corporate world there is hardly any privacy at all. Many businesses openly monitor web usage by its employees. Thus, the privacy of the surfing information is rather very limited at best and may not exist in many cases.
The use of browsers for crawling and transmission of harvested pages to a central database should not make web surfing data less private – only text information from downloaded pages is transmitted and no information about the user. The mere sending of text information from public pages downloaded from the Internet should not be a concern for web surfers. As a matter of fact it is only contractual obligation that prevents traditional web browser from sending private information to its maker or a third party and it is the same way it could be done when using browser-crawler. The maker of a browser-crawler is contractually obligated to prevent what is considered a private data from being sent to a central location. At no circumstance should information that identifies a user such as IP address, email and login names to be sent to a central location. Also other measures need to be taken to safeguard users’ private information. For example, pages of websites that require authentication or are accessed on secure connections should not be harvested since these pages may contain private information.

The use of browser-crawlers in a corporate environment where web surfing by employees is monitored all the time is certainly an option and most likely will not raise many privacy related issues.

2.3 Simulation

2.3.1 Browser-Crawler Application

In order to estimate harvesting capability of the model a simulation of a browser-crawler has been created. The browser-crawler is a client side application written in Java that provides to a user capability to browse the web and at the same time conducts BFS crawl of the web starting with the URL that user points the browser-crawler to. The simulation does not send the harvested pages to a central location for indexing and storage for access by search engine users. Thus, the simulation is not attempting to implement entire search engine but only search engine’s page harvesting agent - browser-crawler. The simulation is used to measure how many pages a browser-crawler can harvest per unit of time. Then harvesting capability of the model can be estimated by using harvesting rate of the browser-crawler and the Internet usage statistics such as number of users browsing the web, average time users spend online, etc.

The simulation of the browser-crawler provides two modes of operation: manual and automated. In manual mode the browser-crawler is driven by a user while he or she surfs the web. The user can point the browser-crawler to a desired page by typing a URL or by clicking a link on the page that has been displayed already. The manual mode essentially provides basic browser capabilities to a user. In automated mode the browser-crawler needs to be given just a start URL and then it simulates web surfing by loading a web page at a time interval that equals an average time that users spend viewing a web page. The purpose of the automated mode is to facilitate data collection.

In either mode the browser-crawler performs BFS crawl in the background using user-provided URL as a starting point. When the starting URL is downloaded it is digested and URLs are extracted from the links on this page. The URLs are added to the list of URLs that need to be crawled. Then first URL from the list is downloaded and processed in the same way to extract URLs to pages that it provides links to. The BFS crawl continues until either all URLs in the list have been processed or is stopped by the user.

The browser-crawler maintains an application log where it writes information such as the times crawl has been started and stopped, current URL being crawled, total number of pages that has been downloaded and digested, etc.

The user interface of the browser-crawler consists of three major parts. The top portion of the browser-crawler’s user interface provides to user capabilities to specify a URL, to download the specified web page, and to select a mode of operation of the browser-crawler.

The central portion of the browser-crawler’s user interface displays the latest web page that has been downloaded and provides capabilities to download a different page by clicking on one of the links that are on the page (if the page has links).

The bottom portion of the browser-crawler’s user interface provides capability to specify location for the applications log file and displays current crawling statistics such as current URL that is being processed, number URLs that has been already processed, and number of URLs that has been extracted from the processed pages and that will be processed if the crawl continues.

In each test environment the browser-crawler has been run multiple times in automated mode to simulate a user surfing the web. First test environment consisted of a relatively old computer with Pentium III processor and dial up connection to the Internet. Second test environment consisted of a newer computer with Pentium IV processor and broadband connection to the Internet.

2.3.2 Data Collection

The browser-crawler has been run to collect data in two different environments in order for results of the simulation to better reflect the real world’s diversity of the Internet users’ environments. The harvesting capability of the browser-crawler is strongly dependent on bandwidth of the Internet connection that the browser-crawler is using to download pages from the web and to a lesser degree processing capabilities of a computer that the browser-crawler runs on. Therefore, the browser-crawler has been run in two distinct test environments that were arbitrary selected as the most representative of what is being used in the real world to surf the web. First test environment consisted of a relatively old computer with Pentium III processor and dial up connection to the Internet. Second test environment consisted of a newer computer with Pentium IV processor and broadband connection to the Internet.

In each test environment the browser-crawler has been run multiple times in automated mode to simulate a user surfing the web.
web. The simulation of a user surfing the web has been achieved by automatically loading a page every 44 seconds, which is approximate average time user spends viewing a web page according to the Internet usage statistics published by Nielsen//NetRatings (available at http://www.nielsen-netratings.com/). Since the content of pages that are being processed determines the size of the pages and that affects the harvesting rate, therefore, the same set of starting URLs has been used for both environments in order to even conditions for the measurement. The starting URLs have been arbitrarily selected to be URLs of well-known sites from various domains. The time browser-crawler ran and crawled the web varied from few minutes to more than hour and a half per run.

The following table summarizes data collected during the experiment. The ‘Test Type’ column specifies what test environment the data corresponds to: either dial up or broadband. The ‘Runs’ column specifies how many times the browser-crawler has been run in the specified environment to generate the data. The ‘Pages Harvested’ column provides average number of pages harvested per run and standard deviation. The ‘Time’ column provides average time in minutes for duration of each run and standard deviation for duration of runs. The ‘Harvesting Rate’ column provides average number of pages harvested per minute and standard deviation for the harvesting rate.

### Table 1. Data Collected With Browser-Crawler

<table>
<thead>
<tr>
<th>Test Type</th>
<th>Runs</th>
<th>Pages Harvested</th>
<th>Time (min)</th>
<th>Harvesting Rate (pgs/min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dialup</td>
<td>22</td>
<td>208.8/171.0</td>
<td>25.0/21.9</td>
<td>9.6/5.0</td>
</tr>
<tr>
<td>Broadband</td>
<td>22</td>
<td>960.4/839.8</td>
<td>21.4/24.6</td>
<td>79.3/74.5</td>
</tr>
</tbody>
</table>

2.3.3 Data Analysis

In order to estimate harvesting capability of the model the Internet usage statistics provided by Nielsen//NetRatings (available at http://www.nielsen-netratings.com/) are being used. According to Nielsen//NetRatings there are total of 819,931,592 users of the Internet by the end of third quarter of 2004 (counting the Internet access from both work and home). Approximately half of the users have broadband access to the Internet (and the share of broadband access is rapidly increasing) while the rest has dial up connection. On average a user spends 51.7 minutes per day using the Internet. Therefore, the maximum number of pages that can be harvested in one day if all users of the Internet used browser-crawlers can be calculated as follows:

Max pages per day = \(819,931,592 \times 51.7 \times (9.6 \times 0.5 + 79.3 \times 0.5)\) = 1,884,256,093,969.48

This number of pages per day is more than 233 times larger than all the pages that Google search engine claims to have indexed to date (8,058,044,651). Of course, the vast majority of these pages are expected to be duplicates of ‘well-known’ pages. The harvesting rate of unique pages will be orders of magnitude less. However, this problem is not unique to the usage of browser-crawlers but is characteristic of crawlers in general. As discussed above this problem may be mitigated by a ‘negotiation’ between browser-crawlers and central location to determine whether a site needs to be crawled or not in order to avoid unnecessary traffic.

The peak harvesting rate of Google is about 100 pages per second [12]. This translates into 8,640,000 pages per day. This number of pages can be harvested by about 3,760 users that utilize browser-crawlers.

3. SUMMARY/CONCLUSIONS

The current model for information retrieval from the web is found to be flawed and inefficient:

- A significant portion of the web cannot be accessed by crawlers and, thus, is not available to retrieve information from
- Crawlers have relatively low harvesting rate that translates into low refresh rate for indexed pages that leads to ‘stale’ pages that are out of date
- Crawlers generate huge amount of traffic that impedes useful communications
- Crawlers require significant resources such as enormous computer farms to harvest and store pages

The new model for information retrieval from the web has been proposed where web browsers are used as main tool to harvest web pages and extract information from the pages. The extracted information is sent by the browser-crawlers to a central location where it is indexed, stored, and is accessible for retrieval by end users. Traditional crawlers are used as an auxiliary tool to harvest pages from portions of the web that are not currently being harvested by browser-crawlers. Given a large user base of browser-crawlers the new model can provide the following benefits:

- The coverage of the web can be significantly improved where browser-crawlers can harvest pages that could not be retrieved by traditional crawlers such as static HTML pages that are not reachable to traditional crawlers, dynamic pages that require user interaction, and pages that are prohibited to crawlers
- The harvesting rate can be significantly improved given a large user base of browser-crawlers that may lead to decreased level of ‘staleness’ of the indexed pages
- The new model provides near real time dynamic view of the usage of the web providing wealth of information about web usage patterns and statistics
- The new model may significantly speed up discovery of new web sites since during deployment of a new web site its web pages are accessed with browser-crawlers to test the site
- The new model may allow reduction in resources required for the process of information retrieval since the tasks of harvesting pages from the web and extracting information from the pages is offloaded to computers hosting browser-crawlers
- The use of browser as a crawler may noticeably improve its user’s web-surfing experience by providing a large cache of harvested pages

The implementation of the new model to retrieve information from the World Wide Web may be impeded by lack of large user base of browser-crawlers due to issues such as concerns about possibilities that browser-crawlers may not provide enough safeguards to prevent private information from being divulged.
However, even if the concept of browser-crawlers is not accepted by the widest audience it may find its niche in large private corporate networks where the new model will provide significant benefits for an environment where choice of browsers can be controlled and privacy is less of an issue.

4. REFERENCES