

Search Engine Update System

Reducing capacity and bandwidth burden of web crawlers

Craig Henderson

August 2008

search@craighenderson.co.uk

<http://www.craighenderson.co.uk/search>

ABSTRACT

The fundamental approach to maintaining a Search Engine's index of the web has not changed since the earliest search engines. The implementations have developed, as have the user interfaces and underlying algorithms, but the core technique of crawling web sites for content remains unchanged. The web is now so large that crawlers cannot get to sites often enough to effectively represent the current content contained on websites. The problem for Web Masters is a challenge large enough that a new market for Search Engine Optimisation has evolved to address the Web Masters' desire to be ranked number one on the top search engines' results page for relevant queries, but they are almost powerless to increase the frequency of visits from crawlers to ensure the most up-to-date content is represented.

This paper assesses the scale of the problem facing organisations that attempt to index or monitor the information published on the World Wide Web and explores the missed opportunities of content that is hidden within inaccessible parts of the web which has become known as the Deep Web.

A system of Site Update Notifications is described as a method for web masters to automatically feed data about content and presentation updates from web servers to those interested. This alternative to the traditional polling of sites distributes the burden and increases efficiency by reducing bandwidth requirements across the internet.

KEYWORDS

Spider, web crawler scalability, optimization, change notification, site update notification

1. INTRODUCTION

Web search engines were developed to feed a user's need to be able to locate relevant information on the world-wide-web, and to do so very quickly.

In January 2008, there were 541,677,360 hosts advertised in the internet DNS, up from 29,670,000 in January 1998; an increase of 1,725% in 10 years [1]. There has been an average of 140 thousand new domain registrations around the world each day between January 1998 and January 2008, and the growth shows no sign of slowing down.

Such a large number of sites on the internet poses significant problems for everyone who uses the internet to publish or discover information;

For Search Engines; How to crawl, index and rank the contents of every page on every site often enough that regularly updated content is contained within the index and is accessible to users

For Web Masters; How to make newly published information findable by interested readers – typically through popular search engines

For End Users; How to find information which is relevant and up-to-date

The fundamental approach to maintaining a Search Engine's index of the web has changed very little since the earliest search engines; a central repository collects information from as many sources as it can and provides a web-based user interface to interrogate the repository using keywords. The implementations have developed, as have the indexing and relevancy ranking algorithms, and user interfaces, but the core technique remains unchanged. This requires heavy investment in the bandwidth and processing power to visit each of

the 140 thousand new sites per day and re-visit existing sites looking for pages that have been updated. Some techniques have been developed to help to reduce the workload of web crawlers (also known as Spiders, Web Wanderers, Robots, Webbots or simply Bots), such as the introduction of The Robots Exclusion Protocol [2] and Sitemaps [3], but these are crude and largely ineffective in solving the larger problem of indexing the Web.

The problem for Web Masters is a challenge large enough that a new market for Search Engine Optimisation (SEO) has evolved to provide documentation, tools and expertise to try to get a high ranking of web content in the major search engines' results.

As the web continues to grow and becomes increasingly dynamic in its content, the ability to maintain an up-to-date index of all sites is less achievable than it has ever been before, and even maintaining an index of a useful size to an increasing customer base is becoming unmanageable. The current solution of adding more bandwidth and capacity for crawling the web is unsustainable in the mid and long term.

2. THE DEEP WEB

In 2001, Michael K. Bergman wrote a white paper *The Deep Web: Surfacing Hidden Value* [4] in which he draws an analogy of a Search Engine dragging a net across the surface of the web, indexing information found on the surface but missing out on the denser and potentially more valuable information buried deep below. This analogy works as well today as it did seven years ago; the Deep Web (also called *Deepnet*, the *Invisible Web*, or the *Hidden Web*) is inaccessible to the mass population using mainstream search engines such as Google, Microsoft Live or Yahoo!

The reasons for this are many. A Search Engine has an enormous job to crawl and index all sites that it knows about because of the volume of information contained within them. Google recently posted a blog entry entitled "*We knew the web was big...*" [5] claiming that it had hit a milestone of one trillion - that's 1,000,000,000,000 - active and unique URLs in its database, although it concedes not to

index them all. Cuil (www.cuil.com), a new player in the search engine space, claims to be the world's biggest search engine with 121 billion pages indexed. This sounds like a lot of pages, but consider that there are 541 million hosts, this averages only 224 pages per host. The numbers are big, and in isolation sound impressive, but the fact is that this is still only the tip of a very large iceberg.

Subscription Sites

A lot of web pages are inaccessible to search engines because their sites restrict access with user accounts. If the spider cannot access the pages, then they will not be able to index them and the content will not be included in the Search Engine's results for relevant queries. While it is important to restrict some content to subscribers, it is also important to recognise that Search Engines provide a means of exposure, too. If protected content was available to search engines such that they could provide summary information to a public user, then this could lead to new subscribers for paid-for content. As it is today, the entire paid-for repository is often inaccessible to any user that is not currently subscribed.

Account-protected content is an increasing trend on the web as businesses recognise the value in maintaining a list of users' email addresses and seek to exploit the large customer base provided by this new medium. As the trend grows, the Deep Web will become deeper, and proportionally less content will be accessible via generic Search Engines.

Online Stores

Online product sales are ever increasing with online sales reaching £26.5bn during the first six months of 2008 in the UK, up 38% on 2007 [6]. Dedicated online stores such as Amazon.com as well as online representations of high street stores such as Mothercare (www.mothercare.co.uk) are therefore keen for their pages to be at the top of Search Engine results for the products that they sell. There is a lot of work to do for these online retailers to structure their site effectively so that the deeper content of items for sale can be accessible to the Search Engine spider. Consumer sales sites

such as eBay.com and Autotrader.co.uk have a bigger challenge still as their sale items are transient; these sites sell one-off items for a limited period, so need to rely on a Search Engine to crawl and index their site very regularly.

Agency sites

Web sites of agency services, such as house sales sites representing estate agents (e.g. realtor.com, rightmove.co.uk) and job vacancy sites representing recruitment agencies (e.g. monster.com) provide sophisticated search user interfaces to enable users to find property, jobs, etc. to suit their needs. The sites typically do not prevent bots from large search engines from accessing their sites, but the content is typically inaccessible to bots that simply navigate the site following href links. Like consumer sales sites, the products (properties, job vacancies, etc.) are transient; specific one-off entries for a limited period, so these sites need a Search Engine to crawl and index their site regularly.

3. ANATOMY OF A SEARCH ENGINE

A Search Engine consists of three core components; the Crawl, the Index and the Query Processor (runtime system). Each component has significant challenges in scale and function.

The Index is an internal system for each search engine provider and is generally bespoke technology that is not subject to wider discussion.

The Query Processor has been the largest area of interest for public research and discussion as this is the consumer facing component of a web search. The search form – which is typically an edit box to accept keywords and a search button – provides consumer access to the search index, and the search results are presented back to the user. Many large search engines that crawl and index the web, such as Google, Microsoft Live Search and Yahoo!, provide third party access to their search indexes via APIs. This has yielded a proliferation of so-called Meta Search Engines which implement the Query Processor but use the Index from other vendors. Meta search engines such as Dogpile, Excite and HotBot aggregate search results from multiple indexes, and present them in a single display.

The Crawl is generally lacking in publically accessible published research. The large search engines are clearly working on advancing the methods of crawling the web, and some research is published [7], but the volume is minimal and does not reflect the size of the problem. Sitemaps was the last major development in web site collaboration for making the crawling of the web more efficient, in June 2005 [3].

There are three problems with the Crawl; Scalability, Frequency and Accessibility. Crawling the web is an enormous task that requires a lot of network bandwidth and processing power. As the number of sites grows, it is increasingly more difficult to crawl the sites regularly enough to access content that is current, and the deep web demonstrates the problem HTTP-client crawlers being unable to penetrate content contained within the Deep Web.

4. SITE UPDATE NOTIFICATIONS

The Site Update Notification system provides an architecture for web sites to supply data to search engines describing changes made to their pages, either in content or in presentation. The distinction is important such that a search engine can determine when an index needs updating or simply an update to the cached representation of a page is enough.

A notification from an update website enables a search engine to re-index targeted pages on a web server only when a change is made, resulting in the search engine crawler and indexing software being more efficient. The crawler’s selection of web sites

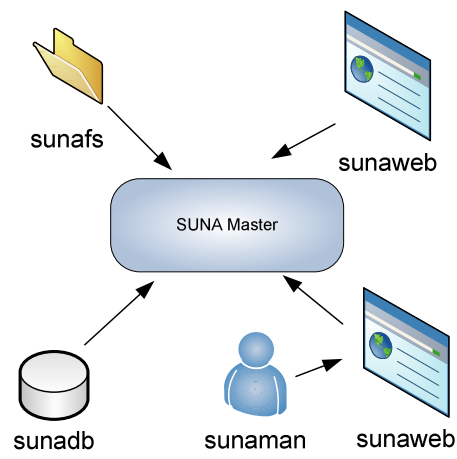


Figure 1 - Architecture of SUNA

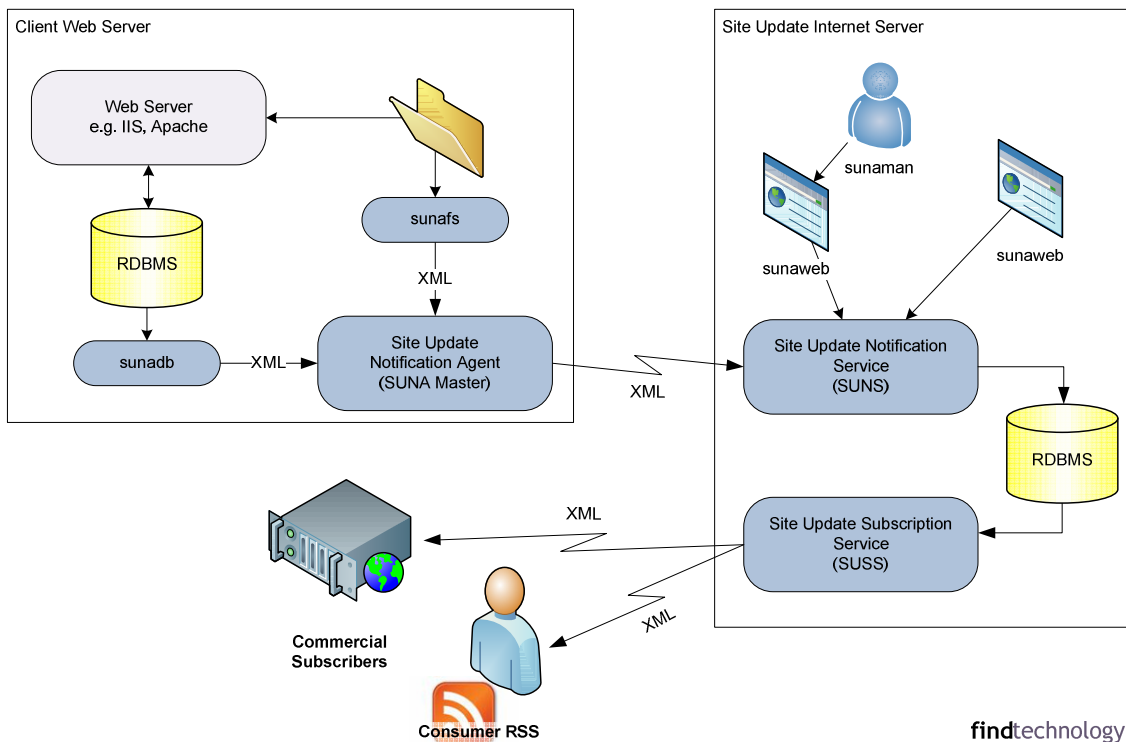


Figure 2 - Site Update Notification Architecture

to visit can avoid those sites known not to have changed since the last visit and therefore make better use of the available bandwidth and processing. Additionally, the bandwidth requirement of client web servers is reduced, and keeps the search engine index up-to-date with dynamically changing content.

Objectives

Current architectures for crawling the web – that of centralised content access over HTTP – have serious shortcomings which need to be overcome to sustain search engine indexes as the size of the web grows, and the importance of Search increases in consumers’ lives.

The Site Update Notifications system presented herein is designed to overcome the current issues with crawler scalability, search engine index currency and restricted access to the Deep Web. The paradigm shift that enables the system to achieve these goals is to distribute the effort of the crawler among the web servers that host the content. In doing so, the function changes from a crawler that

accesses a web site via HTTP requests to a web server agent with enough domain knowledge to create a more accurate record of changes to page content and presentation.

There is evidence to demonstrate that this distribution of effort will be accepted by the web community if the objectives are achieved. It is an obvious fact that search engines want their indexes to be as complete as possible, as well as containing current data, to provide accurate and comprehensive search results. This is evidenced by the massive resources and infrastructures in which the big players have invested to repeatedly crawl websites, and by current research into crawling the Deep Web [7].

The creation and success of the Search Engine Optimisation (SEO) market demonstrates that individual web masters are also willing to invest to increase their site’s position in search engine results for relevant queries. The Site Update Notification System can be used as a valuable SEO tool to feed updates to indexers much more quickly than is currently possible.

Description

A client web server hosts one or more *Site Update Notification Agents* (SUNA's) that monitor for changes in the content of the website. When a change is detected, SUNA collects information about the change and identifies which external web site pages are affected by the change. The SUNA can identify changes that are only to content, or include presentation changes such as updated graphic images or style sheet definitions. This information is transmitted to a *Site Update Notification Service* (SUNS) hosted on an internet server. Connectivity can be established using any communication protocol over TCP/IP, for example, HTTP/S, FTP or any other industry standard or even developed to support proprietary protocols. The data transfer uses an authentication mechanism to prevent unauthorised submission of site update information which could lead to abuse of the service. The information describing changes to web page content across a plurality of client web servers submitted from authenticated SUNA's is collated and stored securely.

Agents (SUNA's)

Site Update Notification Agents typically run on client web servers to locally collate change information. The source of a change in content or presentation can be varied and disparate. The architecture of the client-side agents is designed to provide for a very flexible plug-together component system consisting of one or more source-specific agents along with a master agent – the SUNA Master – that is responsible for collating information from all other local agents and transmitting to the internet hosted SUNS server.

sunafs uses a file system change notification mechanism to monitor for any changes that are made to files in the web server's document root directory and sub directories therein. These directories contain the source files used to produce the website, whether HTML, a client-side script such as JavaScript, or a server-side script repository such as PHP or ASP. *sunafs* will detect changes made to the source files and determine the external URLs which will be affected by the changes.

sunadb monitors changes made in database tables that provide data to be displayed on a web page – such as in a Wiki or Blog system. *sunadb* is fully configurable to support proprietary systems, or one of the pre-configured solutions for popular systems such as Wordpress can be used with minimal customisation.

sunaweb uses proprietary technology and a custom web crawler to monitor changes made to external websites that provide data to be displayed, for example in a mashup system.

sunaman provides a web-based user interface for webmasters to submit manual notifications of content changes and works in conjunction with *sunaweb* to identify the scope of the update. The *sunaman* function is similar to the URL notification form at Yahoo! [8], with the added value of integration with the site update technology of *sunaweb* and SUSS.

By having the agent running on the client web server, the system can respond to real changes made to the source document or dependent data that make up the source document. This is contrary to current techniques that identify changes in the resultant HTML document that is actually delivered from the web server in response to a HTTP GET request. Changes in the client delivered HTML document can be as a result of randomised content on each request, such as time-sensitive advertising, which does not change the user-valued content of the page and should not therefore be considered a reason to download and re-index the page.

Subscription Service

The *Site Update Subscription Service* (SUSS) hosts the provision of site update information to subscribers. The service provides on-demand delivery of change information relating to one or more specific sites, or for all sites, within a given time period. The information is returned to an authenticated subscriber in an XML document. Typical subscribers to the SUSS are those that currently crawl internet web sites for content or benefit in knowing that the content has changed. The largest

of these are web search engine sites, and others include page monitors, bookmark managers, offline browsers and website mirroring.

An alternative subscription service is available to consumers as an RSS feed. This XML document contains a higher level change notification alert to provide an end-user page monitoring service using a standard technology.

5. CRAWLERS' BEHAVIOUR

The adoption of the Site Update Notification System by a search engine company, or other organisation that regularly crawls the web, will inevitably require changes to their Crawl implementations. The modifications are, however, relatively minor in concept as the system can be treated as an enhancement to the existing Crawl method.

The system's purpose is to act as a prompt to subscribers indicating that the content or presentation of specific pages on a site has changed. This prompt indicates that the spider should be sent to crawl those pages, accessing the new or changed information. The current methodology employs proprietary algorithms to schedule the spider to re-crawl sites based on factors such as demonstrated frequency of change (perhaps based on some historical trend analysis), supplied information about the intended frequency of change from the webmaster via sitemaps [3], the popularity of the site - as determined by further proprietary algorithms, or other statistical based guesswork as to the likelihood that a site may have changed since the last crawl. The modification to support the system, then, is in the mechanics of scheduling the crawl away from polling and towards notification, removing the unnecessary effort of re-crawling unchanged pages which in turn frees bandwidth to be available to crawling new content.

An important design goal of the system has been to minimise the impact on spider implementations. The spider retains control of its own schedule based upon its own rules and policies. The system described empowers the spider to be more efficient by providing detailed information about *real* changes that are made to a web page. Rather than a

spider blindly crawling a site, downloading comparatively large volumes of data only to discover that the page is unchanged, the spider is now notified that the page *has* changed, and the spider can make its own decisions on when to act upon that knowledge and re-crawl the page.

6. CASE STUDY

Let us consider an example. `www.cnn.com` is a news site that is constantly updated with current affairs and news stories from around the world. The site's `robots.txt` file is 726 bytes in size, and allows spiders to crawl the news articles.

The homepage is 96,549 bytes at the time of writing, and has 84 dependent files that may affect the content or presentation of the page. These dependent files may include JavaScript source, cascading style sheets, images or other HTML pages contained within iFrames. The total size of the homepage and all its dependent files is 633,045 bytes.

A web spider therefore has to download 618Kb of data the first time it visits the homepage to get a complete picture of the page. Subsequent visits may be optimised by querying the last modified date and only downloading files that have changed, but the last modified HTTP header is not always available for a file and duplicate data is downloaded repeatedly.

More significant is the nature of a change or a difference. Two successive downloads of the same page can generate different HTML documents with exactly the same content and layout. The homepage contains dynamically generated JavaScript that has hard-coded within it the server's current time, seemingly for analytic purposes only. However, because those changes are made to the HTML that is delivered from the server, the agent - web spider or browser - will be required to download a new 96Kb page. With dynamic pages like these, there can be no intelligent downloading by an agent that will optimise the currency of the content held offline and balance the bandwidth and processing power required. The Update Notification mechanism overcomes these problems without the need for major changes being made to the web site. The

web analytics JavaScript generation can remain in place, and will accurately record the timestamp at which the page is downloaded.

The SUSS XML document delivered to subscribers is very much more comprehensive than the consumer RSS feed. The RSS contains detail on which web page has changed, without reference to content versus presentation, or even HTML content versus dependent file changes. Consumers, however, are only interested in the page that has changed. SUSS subscribers receive a more detailed breakdown of changes including individual URLs of dependent files, modified timestamp and size.

Using *sunaman* and *sunaweb* on www.cnn.com produced an XML document for SUSS subscribers of 15,021 bytes for the homepage with 84 dependents and 3,162 bytes for a subsequent crawl where the homepage HTML and fifteen images had changed.

7. CONCLUSION

This paper has described the problem facing organisations that attempt to index or monitor the information published on the World Wide Web. The rate of growth of the number of internet server domains, alongside the frequency of changing content and new content being added, means that continuously crawling web sites to find new and updated content is inefficient and unsustainable.

The analogy of the Deep Web has been discussed and further content accessibility problems with crawling the deep web have been identified.

A system of *Site Update Notifications* was introduced as a method for Web Masters to automatically feed data about content and presentation updates from web servers to subscribers. This alternative to the traditional polling of sites distributes the burden across many servers and increases efficiency by reducing bandwidth requirements across the internet.

Contributions

This paper makes a number of contributions to the field of web crawling.

The first contribution is to address the problem with crawler scalability by distributing the work of the crawler among the web servers that host the content. With the system in place, it is no longer necessary for the crawler to access – and possibly download – every page on a site to determine if it has changed.

The second contribution is in reducing the crawler bandwidth and index load by means of eliminating changes that are generated through the delivery or presentation of the page but do not affect the referenceable information on the page. Such changes were discussed in the case study where a document delivered to a web crawler over HTTP changed between successive downloads although the source files have not changed. Client side agents would not deliver any change notification in situations such as these.

The third contribution is in the ability to reduce the time that a web site's change takes to be available in a search engine's index. News and current affairs sites are updated constantly. In the UK, the national broadcaster the BBC advertises their news web site to be "updated every minute". It is unrealistic to expect a crawler to be able to keep up with this turnaround of content, but a client side agent is more than capable of feeding update information at this frequency.

The fourth contribution addresses the problem crawlers currently face in accessing the Deep Web. While current research centres on trying to penetrate the HTML-form access to deep web content, the Site Update Notification System provides a mechanism to access the Deep Web by working with content publishers.

Finally, the Site Update Notification System provides an infrastructure for agent data aggregation and delivery to subscribers that reduces the bandwidth of individual web sites and provides a mechanism for wider exposure of changing content to customers via RSS feeds.

Benefits

The notification system described has a large number of benefits across all users of the World Wide

Web.

Search Engine Companies and others who crawl the web for content will reduce the workload and bandwidth needs of the crawler and reduce the workload of the indexer by receiving accurate information about web site changes made at source rather than working with changes identified in the HTTP delivered documents. Reducing the workload will enable access to more sites in a shorter timeframe with the same IT resources. Finally, the syndicated search index updates provide a mechanism to keep an index current with fast changing sites such as news and current affairs.

Webmasters and content publishers will improve search result ranking by propagating updates to system subscribers immediately when content is published. Site traffic will be reduced as system subscribers no longer need to indiscriminately crawl the sites downloading unnecessary content, as change notifications are only produced where referenceable information on the site has changed. Site response time will therefore be improved for customers, while reducing bandwidth costs. The maintenance overhead of keeping an accurate and up-to-date sitemap XML file for robots can ultimately be saved.

Hosting service companies can benefit significantly from their customers using the Site Update Notification System. These companies host the websites of their customers and provide a level of server for bandwidth and availability. By providing the system as part of their package to end users, they can

benefit from large savings in reduced bandwidth on the virtual servers.

Finally, the end users – the entire community of web surfers around the world – can benefit from the improved currency in search engine indexes and better site response times caused by the reduction of non-human generated network traffic by world-wide-web spiders.

8. REFERENCES

1. Internet Systems Consortium Internet Domain Survey, <http://www.isc.org/ds>
2. The Robots Exclusion Protocol, robots.txt; <http://www.robotstxt.org/>
3. Google Sitemaps was first introduced in June 2005 <http://googleblog.blogspot.com/2005/06/webmaster-friendly.html> and later joint support from Google, MSN and Yahoo! documented at <http://www.sitemaps.org/>
4. Bergman, Michael K., WHITE PAPER: The Deep Web: Surfacing Hidden Value, Journal of Electronic Publishing 7(1), University of Michigan, August 2001
<http://hdl.handle.net/2027/spo.3336451.0007.104>
5. <http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html>
6. Finch, J., Online sales boom as shoppers desert high street, The Guardian online, 18th July 2008, <http://www.guardian.co.uk/business/2008/jul/18/retail.internet>
7. Google's Deep-Web Crawl, Jayant Madhavan, David Ko, Lucja Kot, Vignesh Ganapathy, Alex Rasmussen, Alon Halevy, Proceedings of the International Conference on Very Large Databases (VLDB), 2008
8. <http://developer.yahoo.com/search/siteexplorer/V1/updateNotification.html>