SEO Parramatta

- News
- SEO Parramatta
- Web Design Parramatta
- Local SEO Parramatta
- Parramatta SEO services
- More
  Parramatta web design agencySearch Engine Optimsation Parramatta Affordable SEO ParramattaCustom web design ParramattaeCommerce web design ParramattaParramatta digital marketingBest SEO agency Parramatta SEO expert ParramattaResponsive web design ParramattaSmall business SEO ParramattaWeb development ParramattaSEO consultant Parramatta Website designers ParramattaSEO company ParramattaWeb design company ParramattaSEO audit Parramatta
- About Us
- Contact Us

SEO maintenance Parramatta —

- Online visibility Parramatta
- SEO maintenance Parramatta
- Creative web design Parramatta
- Parramatta local marketing experts
- Parramatta SEO growth
- Website SEO tune-up Parramatta
- Parramatta web strategy

Take your digital presence further with SEO Parramatta We develop custom strategies aimed at increasing your online visibility, improving search engine rankings, and achieving sustainable growth for your Parramatta-based business

Take your digital presence further with Web Design Parramatta We develop custom strategies aimed at increasing your online visibility, improving search engine rankings, and achieving sustainable growth for your Parramatta-based business

Take your digital presence further with Local SEO Parramatta We develop custom strategies aimed at increasing your online visibility, improving search engine rankings, and achieving sustainable growth for your Parramatta-based business

Effective Web Design Parramatta Sydney.

Creative web design Parramatta

Take your digital presence further with Parramatta SEO services We develop custom strategies aimed at increasing your online visibility, improving search engine rankings, and achieving sustainable growth for your Parramatta-based business

Take your digital presence further with Parramatta web design agency We develop custom strategies aimed at increasing your online visibility, improving search engine rankings, and achieving sustainable growth for your Parramatta-based business

Maximise your business potential with Search Engine Optimisation Parramatta We deliver impactful strategies designed to boost your brand awareness, improve online visibility, and generate a steady flow of qualified leads in Parramatta

Digital Marketing Parramatta .

Custom web design Parramatta - SEO Training Parramatta

1. Local Business SEO Parramatta
2. SEO Maintenance Parramatta
3. Parramatta SEO Agency

Parramatta local marketing experts

Maximise your business potential with Affordable SEO Parramatta We deliver impactful strategies designed to boost your brand awareness, improve online visibility, and generate a steady flow of qualified leads in Parramatta

Experience outstanding online performance through Custom web design Parramatta Our expert team specialises in delivering solutions that improve rankings, drive engagement, and generate valuable leads for consistent business growth in Parramatta

Transform your business growth with eCommerce web design Parramatta Our strategies enhance visibility, attract targeted traffic, and maximise conversions for sustained success Partner with us for measurable digital marketing outcomes today

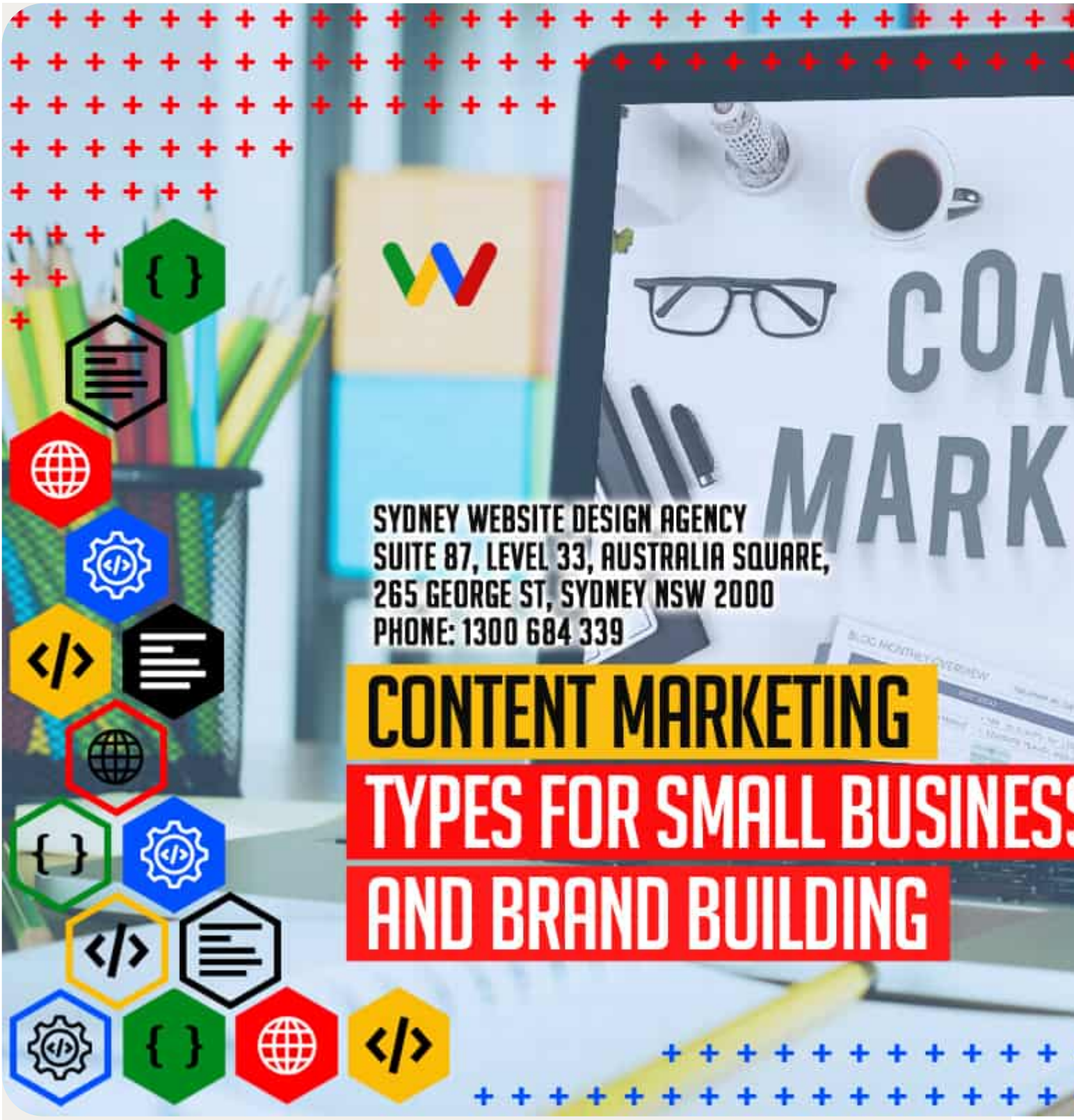
Parramatta SEO growth

Maximise your business potential with Parramatta digital marketing We deliver impactful strategies designed to boost your brand awareness, improve online visibility, and generate a steady flow of qualified leads in Parramatta

Transform your business growth with Best SEO agency Parramatta Our strategies enhance visibility, attract targeted traffic, and maximise conversions for sustained success Partner with us for measurable digital marketing outcomes today

Take your digital presence further with SEO expert Parramatta We develop custom strategies aimed at increasing your online visibility, improving search engine rankings, and achieving sustainable growth for your Parramatta-based business


KEY ADVANTAGE
LOCAL SEO S
28



Website SEO tune-up Parramatta

Take your digital presence further with Responsive web design Parramatta We develop custom strategies aimed at increasing your online visibility, improving search engine rankings, and achieving sustainable growth for your Parramatta-based business

Experience outstanding online performance through Small business SEO Parramatta Our expert team specialises in delivering solutions that improve rankings, drive engagement, and generate valuable leads for consistent business growth in Parramatta

Transform your business growth with Web development Parramatta Our strategies enhance visibility, attract targeted traffic, and maximise conversions for sustained success Partner with us for measurable digital marketing outcomes today

Custom web design Parramatta - SEO Audit Services Parramatta

1. Website Speed Optimisation Parramatta
2. SEO Competitor Analysis Parramatta
3. SEO Content Writing Parramatta

Parramatta web strategy

Transform your business growth with SEO consultant Parramatta Our strategies enhance visibility, attract targeted traffic, and maximise conversions for sustained success Partner with us for measurable digital marketing outcomes today

Take your digital presence further with Website designers Parramatta We develop custom strategies aimed at increasing your online visibility, improving search engine rankings, and achieving sustainable growth for your Parramatta-based business

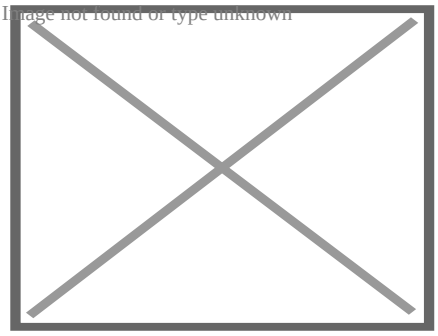
Maximise your business potential with SEO company Parramatta We deliver impactful strategies designed to boost your brand awareness, improve online visibility, and generate a steady flow of qualified leads in Parramatta

Custom web design Parramatta - SEO Training Parramatta

1. SEO Training Parramatta
2. Keyword Research Parramatta
3. SEO Audit Services Parramatta
4. Parramatta Search Visibility

About Web crawler

This article is about the internet bot. For the search engine, see WebCrawler. "Web spider" redirects here; not to be confused with Spider web. "Spiderbot" redirects here. For the video game, see Arac (video game).



Architecture of a Web crawler

A **Web crawler**, sometimes called a **spider** or **spiderbot** and often shortened to **crawler**, is an Internet bot that systematically browses the World Wide Web and that is typically operated by search engines for the purpose of Web indexing (*web spidering*).[1]

Web search engines and some other websites use Web crawling or spidering software to update their web content or indices of other sites' web content. Web crawlers copy pages for processing by a search engine, which indexes the downloaded pages so that users can search more efficiently.

Crawlers consume resources on visited systems and often visit sites unprompted. Issues of schedule, load, and "politeness" come into play when large collections of pages are accessed. Mechanisms exist for public sites not wishing to be crawled to make this known to the crawling agent. For example, including a robots.txt file can request bots to index only parts of a website, or nothing at all.

The number of Internet pages is extremely large; even the largest crawlers fall short of making a complete index. For this reason, search engines struggled to give relevant search results in the early years of the World Wide Web, before 2000. Today, relevant results are given almost instantly.

Crawlers can validate hyperlinks and HTML code. They can also be used for web scraping and data-driven programming.

Nomenclature

[edit]

A web crawler is also known as a *spider*,[2] an *ant*, an *automatic indexer*,[3] or (in the FOAF software context) a *Web scutter*.[4]

Overview

[edit]

A Web crawler starts with a list of URLs to visit. Those first URLs are called the *seeds*. As the crawler visits these URLs, by communicating with web servers that respond to those URLs, it identifies all the hyperlinks in the retrieved web pages and adds them to the list of URLs to visit, called the *crawl frontier*. URLs from the frontier are recursively visited according to a set of policies. If the crawler is performing archiving of websites (or web archiving), it copies and saves the information as it goes. The archives are usually stored in such a way they can be viewed, read and navigated as if they were on the live web, but are preserved as 'snapshots'.[5]

The archive is known as the *repository* and is designed to store and manage the collection of web pages. The repository only stores HTML pages and these pages are stored as distinct files. A repository is similar to any other system that stores data, like a modern-day database. The only difference is that a repository does not need all the functionality offered by a database system. The repository stores the most recent version of the web page retrieved by the crawler.[*citation needed*]

The large volume implies the crawler can only download a limited number of the Web pages within a given time, so it needs to prioritize its downloads. The high rate of change can imply the pages might have already been updated or even deleted.

The number of possible URLs crawled being generated by server-side software has also made it difficult for web crawlers to avoid retrieving duplicate content. Endless combinations of HTTP GET (URL-based) parameters exist, of which only a small selection will actually return unique content. For example, a simple online photo gallery may offer three options to users, as specified through HTTP GET parameters in the URL. If there exist four ways to sort images, three choices of thumbnail size, two file formats, and an option to disable user-provided content, then the same set of content can be accessed with 48 different URLs, all of which may be linked on the site. This mathematical combination creates a problem for crawlers, as they must sort through endless combinations of relatively minor scripted changes in order to retrieve unique content.

As Edwards *et al.* noted, "Given that the bandwidth for conducting crawls is neither infinite nor free, it is becoming essential to crawl the Web in not only a scalable, but efficient way, if some reasonable measure of quality or freshness is to be maintained."[6] A crawler must carefully choose at each step which pages to visit next.

Crawling policy

[edit]

The behavior of a Web crawler is the outcome of a combination of policies:[7]

- a *selection policy* which states the pages to download,
- a *re-visit policy* which states when to check for changes to the pages,
- a *politeness policy* that states how to avoid overloading websites.
- a *parallelization policy* that states how to coordinate distributed web crawlers.

Selection policy

[edit]

Given the current size of the Web, even large search engines cover only a portion of the publicly available part. A 2009 study showed even large-scale search engines index no more than 40–70% of the indexable Web;[8] a previous study by Steve Lawrence and Lee Giles showed that no search engine indexed more than 16% of the Web in 1999.[9] As a crawler always downloads just a fraction of the Web pages, it is highly desirable for the downloaded fraction to contain the most relevant pages and not just a random sample of the Web.

This requires a metric of importance for prioritizing Web pages. The importance of a page is a function of its intrinsic quality, its popularity in terms of links or visits, and even of its URL (the latter is the case of vertical search engines restricted to a single top-level domain, or search engines restricted to a fixed Web site). Designing a good selection policy has an added difficulty: it must work with partial information, as the complete set of Web pages is not known during crawling.

Junghoo Cho *et al.* made the first study on policies for crawling scheduling. Their data set was a 180,000-pages crawl from the stanford.edu domain, in which a crawling simulation was done with different strategies.[10] The ordering metrics tested were breadth-first, backlink count and partial PageRank calculations. One of the conclusions was that if the crawler wants to download pages with high Pagerank early during the crawling process, then the partial Pagerank strategy is the better, followed by breadth-first and backlink-count. However, these results are for just a single domain. Cho also wrote his PhD dissertation at Stanford on web crawling.[11]

Najork and Wiener performed an actual crawl on 328 million pages, using breadth-first ordering.[12] They found that a breadth-first crawl captures pages with high Pagerank early in the crawl (but they did not compare this strategy against other strategies). The explanation given by the authors for this result is that "the most important pages have many links to them from numerous hosts, and those links will be found early, regardless of on which host or page the crawl originates."

Abiteboul designed a crawling strategy based on an algorithm called OPIC (On-line Page Importance Computation).[13] In OPIC, each page is given an initial sum of "cash" that is distributed equally among the pages it points to. It is similar to a PageRank computation, but it is faster and is only done in one step. An OPIC-driven crawler downloads first the pages in the crawling frontier with higher amounts of "cash". Experiments were carried in a 100,000-pages synthetic graph with a power-law distribution of in-links. However, there was no comparison with other strategies nor experiments in the real Web.

Boldi *et al.* used simulation on subsets of the Web of 40 million pages from the .it domain and 100 million pages from the WebBase crawl, testing breadth-first against depth-first, random ordering

and an omniscient strategy. The comparison was based on how well PageRank computed on a partial crawl approximates the true PageRank value. Some visits that accumulate PageRank very quickly (most notably, breadth-first and the omniscient visit) provide very poor progressive approximations.[14][15]

Baeza-Yates *et al.* used simulation on two subsets of the Web of 3 million pages from the .gr and .cl domain, testing several crawling strategies.[16] They showed that both the OPIC strategy and a strategy that uses the length of the per-site queues are better than breadth-first crawling, and that it is also very effective to use a previous crawl, when it is available, to guide the current one.

Daneshpajouh *et al.* designed a community based algorithm for discovering good seeds.[17] Their method crawls web pages with high PageRank from different communities in less iteration in comparison with crawl starting from random seeds. One can extract good seed from a previously-crawled-Web graph using this new method. Using these seeds, a new crawl can be very effective.

Restricting followed links

[edit]

A crawler may only want to seek out HTML pages and avoid all other MIME types. In order to request only HTML resources, a crawler may make an HTTP HEAD request to determine a Web resource's MIME type before requesting the entire resource with a GET request. To avoid making numerous HEAD requests, a crawler may examine the URL and only request a resource if the URL ends with certain characters such as .html, .htm, .asp, .aspx, .php, .jsp, .jspx or a slash. This strategy may cause numerous HTML Web resources to be unintentionally skipped.

Some crawlers may also avoid requesting any resources that have a "?" in them (are dynamically produced) in order to avoid spider traps that may cause the crawler to download an infinite number of URLs from a Web site. This strategy is unreliable if the site uses URL rewriting to simplify its URLs.

URL normalization

[edit]
Main article: URL normalization

Crawlers usually perform some type of URL normalization in order to avoid crawling the same resource more than once. The term *URL normalization*, also called *URL canonicalization*, refers to the process of modifying and standardizing a URL in a consistent manner. There are several types of normalization that may be performed including conversion of URLs to lowercase, removal of "." and ".." segments, and adding trailing slashes to the non-empty path component.[18]

Path-ascending crawling

[edit]

Some crawlers intend to download/upload as many resources as possible from a particular web site. So *path-ascending crawler* was introduced that would ascend to every path in each URL that it intends to crawl.[19] For example, when given a seed URL of http://llama.org/hamster/monkey/page.html, it will attempt to crawl /hamster/monkey/, /hamster/, and /. Cothey found that a path-ascending crawler was very effective in finding isolated resources, or resources for which no inbound link would have been found in regular crawling.

Focused crawling

[edit]
Main article: Focused crawler

The importance of a page for a crawler can also be expressed as a function of the similarity of a page to a given query. Web crawlers that attempt to download pages that are similar to each other are called **focused crawler** or **topical crawlers**. The concepts of topical and focused crawling were first introduced by Filippo Menczer[20][21] and by Soumen Chakrabarti *et al.*[22]

The main problem in focused crawling is that in the context of a Web crawler, we would like to be able to predict the similarity of the text of a given page to the query before actually downloading the page. A possible predictor is the anchor text of links; this was the approach taken by Pinkerton[23] in the first web crawler of the early days of the Web. Diligenti *et al.*[24] propose using the complete content of the pages already visited to infer the similarity between the driving query and the pages that have not been visited yet. The performance of a focused crawling depends mostly on the richness of links in the specific topic being searched, and a focused crawling usually relies on a general Web search engine for providing starting points.

Academic focused crawler

[edit]

An example of the focused crawlers are academic crawlers, which crawls free-access academic related documents, such as the *citeseerxbot*, which is the crawler of CiteSeer^X search engine. Other academic search engines are Google Scholar and Microsoft Academic Search etc. Because most academic papers are published in PDF formats, such kind of crawler is particularly interested in crawling PDF, PostScript files, Microsoft Word including their zipped formats. Because of this, general open-source crawlers, such as Heritrix, must be customized to filter out other MIME types, or a middleware is used to extract these documents out and import them to the focused crawl database and repository.[25] Identifying whether these documents are academic or not is

challenging and can add a significant overhead to the crawling process, so this is performed as a post crawling process using machine learning or regular expression algorithms. These academic documents are usually obtained from home pages of faculties and students or from publication page of research institutes. Because academic documents make up only a small fraction of all web pages, a good seed selection is important in boosting the efficiencies of these web crawlers.[26] Other academic crawlers may download plain text and HTML files, that contains metadata of academic papers, such as titles, papers, and abstracts. This increases the overall number of papers, but a significant fraction may not provide free PDF downloads.

Semantic focused crawler

[edit]

Another type of focused crawlers is semantic focused crawler, which makes use of domain ontologies to represent topical maps and link Web pages with relevant ontological concepts for the selection and categorization purposes.[27] In addition, ontologies can be automatically updated in the crawling process. Dong et al.[28] introduced such an ontology-learning-based crawler using a support-vector machine to update the content of ontological concepts when crawling Web pages.

Re-visit policy

[edit]

The Web has a very dynamic nature, and crawling a fraction of the Web can take weeks or months. By the time a Web crawler has finished its crawl, many events could have happened, including creations, updates, and deletions.

From the search engine's point of view, there is a cost associated with not detecting an event, and thus having an outdated copy of a resource. The most-used cost functions are freshness and age.[29]

Freshness: This is a binary measure that indicates whether the local copy is accurate or not. The freshness of a page *p* in the repository at time *t* is defined as:

{\displaystyle F_{p}(t)={\begin{cases}1&{\rm {if}}~p~{\rm {~is~equal~to~the~local~copy~at~time}}~t\

Image not found or type unknown

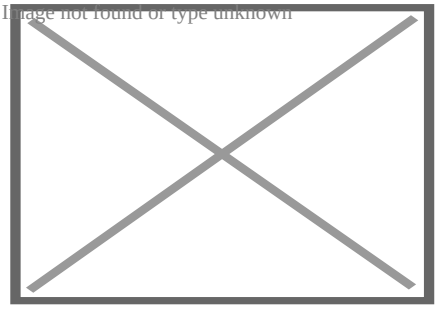
Age: This is a measure that indicates how outdated the local copy is. The age of a page *p* in the repository, at time *t* is defined as:

{\displaystyle A_{p}(t)={\begin{cases}0&{\rm {if}}~p~{\rm {~is~not~modified~at~time}}~t\\t-{\rm {mod

Image not found or type unknown

Coffman *et al.* worked with a definition of the objective of a Web crawler that is equivalent to freshness, but use a different wording: they propose that a crawler must minimize the fraction of time pages remain outdated. They also noted that the problem of Web crawling can be modeled as a multiple-queue, single-server polling system, on which the Web crawler is the server and the Web sites are the queues. Page modifications are the arrival of the customers, and switch-over times are the interval between page accesses to a single Web site. Under this model, mean waiting time for a customer in the polling system is equivalent to the average age for the Web crawler.[30]

The objective of the crawler is to keep the average freshness of pages in its collection as high as possible, or to keep the average age of pages as low as possible. These objectives are not equivalent: in the first case, the crawler is just concerned with how many pages are outdated, while in the second case, the crawler is concerned with how old the local copies of pages are.



Evolution of Freshness and Age in a web crawler

Two simple re-visiting policies were studied by Cho and Garcia-Molina:[31]

- Uniform policy: This involves re-visiting all pages in the collection with the same frequency, regardless of their rates of change.
- Proportional policy: This involves re-visiting more often the pages that change more frequently. The visiting frequency is directly proportional to the (estimated) change frequency.

In both cases, the repeated crawling order of pages can be done either in a random or a fixed order.

Cho and Garcia-Molina proved the surprising result that, in terms of average freshness, the uniform policy outperforms the proportional policy in both a simulated Web and a real Web crawl. Intuitively, the reasoning is that, as web crawlers have a limit to how many pages they can crawl in a given time frame, (1) they will allocate too many new crawls to rapidly changing pages at the expense of less frequently updating pages, and (2) the freshness of rapidly changing pages lasts for shorter period than that of less frequently changing pages. In other words, a proportional policy allocates more resources to crawling frequently updating pages, but experiences less overall freshness time from them.

To improve freshness, the crawler should penalize the elements that change too often.[32] The optimal re-visiting policy is neither the uniform policy nor the proportional policy. The optimal method for keeping average freshness high includes ignoring the pages that change too often, and the optimal for keeping average age low is to use access frequencies that monotonically (and sub-

linearly) increase with the rate of change of each page. In both cases, the optimal is closer to the uniform policy than to the proportional policy: as Coffman *et al.* note, "in order to minimize the expected obsolescence time, the accesses to any particular page should be kept as evenly spaced as possible".[30] Explicit formulas for the re-visit policy are not attainable in general, but they are obtained numerically, as they depend on the distribution of page changes. Cho and Garcia-Molina show that the exponential distribution is a good fit for describing page changes,[32] while Ipeirotis *et al.* show how to use statistical tools to discover parameters that affect this distribution.[33] The re-visiting policies considered here regard all pages as homogeneous in terms of quality ("all pages on the Web are worth the same"), something that is not a realistic scenario, so further information about the Web page quality should be included to achieve a better crawling policy.

Politeness policy

[edit]

Crawlers can retrieve data much quicker and in greater depth than human searchers, so they can have a crippling impact on the performance of a site. If a single crawler is performing multiple requests per second and/or downloading large files, a server can have a hard time keeping up with requests from multiple crawlers.

As noted by Koster, the use of Web crawlers is useful for a number of tasks, but comes with a price for the general community.[34] The costs of using Web crawlers include:

- network resources, as crawlers require considerable bandwidth and operate with a high degree of parallelism during a long period of time;
- server overload, especially if the frequency of accesses to a given server is too high;
- poorly written crawlers, which can crash servers or routers, or which download pages they cannot handle; and
- personal crawlers that, if deployed by too many users, can disrupt networks and Web servers.

A partial solution to these problems is the robots exclusion protocol, also known as the robots.txt protocol that is a standard for administrators to indicate which parts of their Web servers should not be accessed by crawlers.[35] This standard does not include a suggestion for the interval of visits to the same server, even though this interval is the most effective way of avoiding server overload. Recently commercial search engines like Google, Ask Jeeves, MSN and Yahoo! Search are able to use an extra "Crawl-delay:" parameter in the robots.txt file to indicate the number of seconds to delay between requests.

The first proposed interval between successive pageloads was 60 seconds.[36] However, if pages were downloaded at this rate from a website with more than 100,000 pages over a perfect connection with zero latency and infinite bandwidth, it would take more than 2 months to download only that entire Web site; also, only a fraction of the resources from that Web server would be used.

Cho uses 10 seconds as an interval for accesses,[31] and the WIRE crawler uses 15 seconds as the default.[37] The MercatorWeb crawler follows an adaptive politeness policy: if it took t seconds to download a document from a given server, the crawler waits for $10t$ seconds before downloading the next page.[38] Dill *et al.* use 1 second.[39]

For those using Web crawlers for research purposes, a more detailed cost-benefit analysis is needed and ethical considerations should be taken into account when deciding where to crawl and how fast to crawl.[40]

Anecdotal evidence from access logs shows that access intervals from known crawlers vary between 20 seconds and 3–4 minutes. It is worth noticing that even when being very polite, and taking all the safeguards to avoid overloading Web servers, some complaints from Web server administrators are received. Sergey Brin and Larry Page noted in 1998, "... running a crawler which connects to more than half a million servers ... generates a fair amount of e-mail and phone calls. Because of the vast number of people coming on line, there are always those who do not know what a crawler is, because this is the first one they have seen."[41]

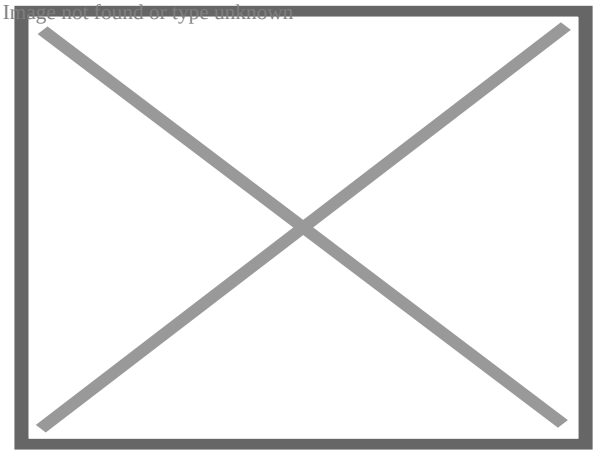
Parallelization policy

[edit]
Main article: Distributed web crawling

A parallel crawler is a crawler that runs multiple processes in parallel. The goal is to maximize the download rate while minimizing the overhead from parallelization and to avoid repeated downloads of the same page. To avoid downloading the same page more than once, the crawling system requires a policy for assigning the new URLs discovered during the crawling process, as the same URL can be found by two different crawling processes.

Architectures

[edit]



High-level architecture of a standard Web crawler

A crawler must not only have a good crawling strategy, as noted in the previous sections, but it should also have a highly optimized architecture.

Shkapenyuk and Suel noted that:[42]

> While it is fairly easy to build a slow crawler that downloads a few pages per second for a short period of time, building a high-performance system that can download hundreds of millions of pages over several weeks presents a number of challenges in system design, I/O and network efficiency, and robustness and manageability.

Web crawlers are a central part of search engines, and details on their algorithms and architecture are kept as business secrets. When crawler designs are published, there is often an important lack of detail that prevents others from reproducing the work. There are also emerging concerns about "search engine spamming", which prevent major search engines from publishing their ranking algorithms.

Security

[edit]

While most of the website owners are keen to have their pages indexed as broadly as possible to have strong presence in search engines, web crawling can also have unintended consequences and lead to a compromise or data breach if a search engine indexes resources that should not be publicly available, or pages revealing potentially vulnerable versions of software.

Main article: Google hacking

Apart from standard web application security recommendations website owners can reduce their exposure to opportunistic hacking by only allowing search engines to index the public parts of their websites (with robots.txt) and explicitly blocking them from indexing transactional parts (login pages, private pages, etc.).

Crawler identification

[edit]

Web crawlers typically identify themselves to a Web server by using the User-agent field of an HTTP request. Web site administrators typically examine their Web servers' log and use the user agent field to determine which crawlers have visited the web server and how often. The user agent field may include a URL where the Web site administrator may find out more information about the crawler. Examining Web server log is tedious task, and therefore some administrators use tools to identify, track and verify Web crawlers. Spambots and other malicious Web crawlers are unlikely to place identifying information in the user agent field, or they may mask their identity as a browser or other well-known crawler.

Web site administrators prefer Web crawlers to identify themselves so that they can contact the owner if needed. In some cases, crawlers may be accidentally trapped in a crawler trap or they may be overloading a Web server with requests, and the owner needs to stop the crawler. Identification is also useful for administrators that are interested in knowing when they may expect their Web pages to be indexed by a particular search engine.

Crawling the deep web

[edit]

A vast amount of web pages lie in the deep or invisible web.[43] These pages are typically only accessible by submitting queries to a database, and regular crawlers are unable to find these pages if there are no links that point to them. Google's Sitemaps protocol and mod oai[44] are intended to allow discovery of these deep-Web resources.

Deep web crawling also multiplies the number of web links to be crawled. Some crawlers only take some of the URLs in <a href="URL"> form. In some cases, such as the Googlebot, Web crawling is done on all text contained inside the hypertext content, tags, or text.

Strategic approaches may be taken to target deep Web content. With a technique called screen scraping, specialized software may be customized to automatically and repeatedly query a given Web form with the intention of aggregating the resulting data. Such software can be used to span multiple Web forms across multiple Websites. Data extracted from the results of one Web form submission can be taken and applied as input to another Web form thus establishing continuity across the Deep Web in a way not possible with traditional web crawlers.[45]

Pages built on AJAX are among those causing problems to web crawlers. Google has proposed a format of AJAX calls that their bot can recognize and index.[46]

Visual vs programmatic crawlers

[edit]

There are a number of "visual web scraper/crawler" products available on the web which will crawl pages and structure data into columns and rows based on the users requirements. One of the main difference between a classic and a visual crawler is the level of programming ability required to set up a crawler. The latest generation of "visual scrapers" remove the majority of the programming skill needed to be able to program and start a crawl to scrape web data.

The visual scraping/crawling method relies on the user "teaching" a piece of crawler technology, which then follows patterns in semi-structured data sources. The dominant method for teaching a visual crawler is by highlighting data in a browser and training columns and rows. While the technology is not new, for example it was the basis of Needlebase which has been bought by Google (as part of a larger acquisition of ITA Labs[47]), there is continued growth and investment in this area by investors and end-users.[*citation needed*]

List of web crawlers

[edit]
Further information: List of search engine software

The following is a list of published crawler architectures for general-purpose crawlers (excluding focused web crawlers), with a brief description that includes the names given to the different components and outstanding features:

Historical web crawlers

[edit]

- WolfBot was a massively multi threaded crawler built in 2001 by Mani Singh a Civil Engineering graduate from the University of California at Davis.
- World Wide Web Worm was a crawler used to build a simple index of document titles and URLs. The index could be searched by using the grep Unix command.
- Yahoo! Slurp was the name of the Yahoo! Search crawler until Yahoo! contracted with Microsoft to use Bingbot instead.

In-house web crawlers

[edit]

- Applebot is Apple's web crawler. It supports Siri and other products.[48]
- Bingbot is the name of Microsoft's Bing webcrawler. It replaced *Msnbot*.
- Baiduspider is Baidu's web crawler.
- DuckDuckBot is DuckDuckGo's web crawler.
- Googlebot is described in some detail, but the reference is only about an early version of its architecture, which was written in C++ and Python. The crawler was integrated with the indexing process, because text parsing was done for full-text indexing and also for URL extraction. There is a URL server that sends lists of URLs to be fetched by several crawling processes. During parsing, the URLs found were passed to a URL server that checked if the URL have been previously seen. If not, the URL was added to the queue of the URL server.
- WebCrawler was used to build the first publicly available full-text index of a subset of the Web. It was based on lib-WWW to download pages, and another program to parse and order URLs for breadth-first exploration of the Web graph. It also included a real-time crawler that followed links based on the similarity of the anchor text with the provided query.
- WebFountain is a distributed, modular crawler similar to Mercator but written in C++.
- Xenon is a web crawler used by government tax authorities to detect fraud.[49][50]

Commercial web crawlers

[edit]

The following web crawlers are available, for a price::

- Diffbot - programmatic general web crawler, available as an API
- SortSite - crawler for analyzing websites, available for Windows and Mac OS
- Swiftbot - Swiftype's web crawler, available as software as a service
- Aleph Search - web crawler allowing massive collection with high scalability

Open-source crawlers

[edit]

- Apache Nutch is a highly extensible and scalable web crawler written in Java and released under an Apache License. It is based on Apache Hadoop and can be used with Apache Solr or Elasticsearch.
- Grub was an open source distributed search crawler that Wikia Search used to crawl the web.
- Heritrix is the Internet Archive's archival-quality crawler, designed for archiving periodic snapshots of a large portion of the Web. It was written in Java.
- ht://Dig includes a Web crawler in its indexing engine.
- HTTrack uses a Web crawler to create a mirror of a web site for off-line viewing. It is written in C and released under the GPL.
- Norconex Web Crawler is a highly extensible Web Crawler written in Java and released under an Apache License. It can be used with many repositories such as Apache Solr, Elasticsearch, Microsoft Azure Cognitive Search, Amazon CloudSearch and more.
- mnoGoSearch is a crawler, indexer and a search engine written in C and licensed under the GPL (*NIX machines only)
- Open Search Server is a search engine and web crawler software release under the GPL.
- Scrapy, an open source webcrawler framework, written in python (licensed under BSD).
- Seeks, a free distributed search engine (licensed under AGPL).
- StormCrawler, a collection of resources for building low-latency, scalable web crawlers on Apache Storm (Apache License).
- tkWWW Robot, a crawler based on the tkWWW web browser (licensed under GPL).
- GNU Wget is a command-line-operated crawler written in C and released under the GPL. It is typically used to mirror Web and FTP sites.
- YaCy, a free distributed search engine, built on principles of peer-to-peer networks (licensed under GPL).

See also

[edit]

- Automatic indexing
- Gnutella crawler
- Web archiving
- Webgraph

- Website mirroring software
- Search Engine Scraping
- Web scraping

References

[edit]

1. ^ *"Web Crawlers: Browsing the Web". Archived from the original on 6 December 2021.*
2. ^ *Spetka, Scott. "The TkWWW Robot: Beyond Browsing". NCSA. Archived from the original on 3 September 2004. Retrieved 21 November 2010.*
3. ^ *Kobayashi, M. & Takeda, K. (2000). "Information retrieval on the web". ACM Computing Surveys. **32** (2): 144–173. CiteSeerX 10.1.1.126.6094. doi:10.1145/358923.358934. S2CID 3710903.*
4. ^ See definition of scutter on FOAF Project's wiki Archived 13 December 2009 at the Wayback Machine
5. ^ *Masanès, Julien (15 February 2007). Web Archiving. Springer. p. 1. ISBN 978-3-54046332-0. Retrieved 24 April 2014.*
6. ^ *Edwards, J.; McCurley, K. S.; and Tomlin, J. A. (2001). "An adaptive model for optimizing performance of an incremental web crawler". Proceedings of the 10th international conference on World Wide Web. pp. 106–113. CiteSeerX 10.1.1.1018.1506. doi: 10.1145/371920.371960. ISBN 978-1581133486. S2CID 10316730. Archived from the original on 25 June 2014. Retrieved 25 January 2007.*{{cite book}}: CS1 maint: multiple names: authors list (link)
7. ^ *Castillo, Carlos (2004). Effective Web Crawling (PhD thesis). University of Chile. Retrieved 3 August 2010.*
8. ^ *Gulls, A.; A. Signori (2005). "The indexable web is more than 11.5 billion pages". Special interest tracks and posters of the 14th international conference on World Wide Web. ACM Press. pp. 902–903. doi:10.1145/1062745.1062789.*
9. ^ *Lawrence, Steve; C. Lee Giles (8 July 1999). "Accessibility of information on the web". Nature. **400** (6740): 107–9. Bibcode:1999Natur.400..107L. doi:10.1038/21987. PMID 10428673. S2CID 4347646.*
10. ^ *Cho, J.; Garcia-Molina, H.; Page, L. (April 1998). "Efficient Crawling Through URL Ordering". Seventh International World-Wide Web Conference. Brisbane, Australia. doi: 10.1142/3725. ISBN 978-981-02-3400-3. Retrieved 23 March 2009.*
11. ^ Cho, Junghoo, "Crawling the Web: Discovery and Maintenance of a Large-Scale Web Data", PhD dissertation, Department of Computer Science, Stanford University, November 2001.
12. ^ Najork, Marc and Janet L. Wiener. "Breadth-first crawling yields high-quality pages". Archived 24 December 2017 at the Wayback Machine In: *Proceedings of the Tenth Conference on World Wide Web*, pages 114–118, Hong Kong, May 2001. Elsevier Science.
13. ^ *Abiteboul, Serge; Mihai Preda; Gregory Cobena (2003). "Adaptive on-line page importance computation". Proceedings of the 12th international conference on World Wide Web. Budapest, Hungary: ACM. pp. 280–290. doi:10.1145/775152.775192. ISBN 1-58113-680-3. Retrieved 22 March 2009.*

14. ^ *Boldi, Paolo; Bruno Codenotti; Massimo Santini; Sebastiano Vigna (2004). "UbiCrawler: a scalable fully distributed Web crawler" (PDF). Software: Practice and Experience. **34** (8): 711–726. CiteSeerX 10.1.1.2.5538. doi:10.1002/spe.587. S2CID 325714. Archived from the original (PDF) on 20 March 2009. Retrieved 23 March 2009.*
15. ^ *Boldi, Paolo; Massimo Santini; Sebastiano Vigna (2004). "Do Your Worst to Make the Best: Paradoxical Effects in PageRank Incremental Computations" (PDF). Algorithms and Models for the Web-Graph. Lecture Notes in Computer Science. Vol. 3243. pp. 168–180. doi: 10.1007/978-3-540-30216-2_14. ISBN 978-3-540-23427-2. Archived from the original (PDF) on 1 October 2005. Retrieved 23 March 2009.*
16. ^ Baeza-Yates, R.; Castillo, C.; Marin, M. and Rodriguez, A. (2005). "Crawling a Country: Better Strategies than Breadth-First for Web Page Ordering." In: *Proceedings of the Industrial and Practical Experience track of the 14th conference on World Wide Web*, pages 864–872, Chiba, Japan. ACM Press.
17. ^ Shervin Daneshpajouh, Mojtaba Mohammadi Nasiri, Mohammad Ghodsi, A Fast Community Based Algorithm for Generating Crawler Seeds Set. In: *Proceedings of 4th International Conference on Web Information Systems and Technologies* (*Webist*-2008), Funchal, Portugal, May 2008.
18. ^ *Pant, Gautam; Srinivasan, Padmini; Menczer, Filippo (2004). "Crawling the Web" (PDF). In Levene, Mark; Poulovassilis, Alexandra (eds.). Web Dynamics: Adapting to Change in Content, Size, Topology and Use. Springer. pp. 153–178. ISBN 978-3-540-40676-1. Archived from the original (PDF) on 20 March 2009. Retrieved 9 May 2006.*
19. ^ *Cothey, Viv (2004). "Web-crawling reliability" (PDF). Journal of the American Society for Information Science and Technology. **55** (14): 1228–1238. CiteSeerX 10.1.1.117.185. doi: 10.1002/asi.20078.*
20. ^ Menczer, F. (1997). ARACHNID: Adaptive Retrieval Agents Choosing Heuristic Neighborhoods for Information Discovery Archived 21 December 2012 at the Wayback Machine. In D. Fisher, ed., Machine Learning: Proceedings of the 14th International Conference (ICML97). Morgan Kaufmann
21. ^ Menczer, F. and Belew, R.K. (1998). Adaptive Information Agents in Distributed Textual Environments Archived 21 December 2012 at the Wayback Machine. In K. Sycara and M. Wooldridge (eds.) Proc. 2nd Intl. Conf. on Autonomous Agents (Agents '98). ACM Press
22. ^ *Chakrabarti, Soumen; Van Den Berg, Martin; Dom, Byron (1999). "Focused crawling: A new approach to topic-specific Web resource discovery" (PDF). Computer Networks. **31** (11–16): 1623–1640. doi:10.1016/s1389-1286(99)00052-3. Archived from the original (PDF) on 17 March 2004.*
23. ^ Pinkerton, B. (1994). Finding what people want: Experiences with the WebCrawler. In Proceedings of the First World Wide Web Conference, Geneva, Switzerland.
24. ^ Diligenti, M., Coetzee, F., Lawrence, S., Giles, C. L., and Gori, M. (2000). Focused crawling using context graphs. In Proceedings of 26th International Conference on Very Large Databases (VLDB), pages 527-534, Cairo, Egypt.
25. ^ *Wu, Jian; Teregowda, Pradeep; Khabsa, Madian; Carman, Stephen; Jordan, Douglas; San Pedro Wandelmer, Jose; Lu, Xin; Mitra, Prasenjit; Giles, C. Lee (2012). "Web crawler middleware for search engine digital libraries". Proceedings of the twelfth international workshop on Web information and data management - WIDM '12. p. 57. doi: 10.1145/2389936.2389949. ISBN 9781450317207. S2CID 18513666.*

26. ^ *Wu, Jian; Teregowda, Pradeep; Ramírez, Juan Pablo Fernández; Mitra, Prasenjit; Zheng, Shuyi; Giles, C. Lee (2012). "The evolution of a crawling strategy for an academic document search engine". Proceedings of the 3rd Annual ACM Web Science Conference on - Web Sci '12. pp. 340–343. doi:10.1145/2380718.2380762. ISBN 9781450312288. S2CID 16718130.*
27. ^ *Dong, Hai; Hussain, Farookh Khadeer; Chang, Elizabeth (2009). "State of the Art in Semantic Focused Crawlers". Computational Science and Its Applications – ICCSA 2009. Lecture Notes in Computer Science. Vol. 5593. pp. 910–924. doi:10.1007/978-3-642-02457-3_74. hdl:20.500.11937/48288. ISBN 978-3-642-02456-6.*
28. ^ *Dong, Hai; Hussain, Farookh Khadeer (2013). "SOF: A semi-supervised ontology-learning-based focused crawler". Concurrency and Computation: Practice and Experience. **25** (12): 1755–1770. doi:10.1002/cpe.2980. S2CID 205690364.*
29. ^ *Junghoo Cho; Hector Garcia-Molina (2000). "Synchronizing a database to improve freshness" (PDF). Proceedings of the 2000 ACM SIGMOD international conference on Management of data. Dallas, Texas, United States: ACM. pp. 117–128. doi:10.1145/342009.335391. ISBN 1-58113-217-4. Retrieved 23 March 2009.*
30. ^ ***a b*** *E. G. Coffman Jr; Zhen Liu; Richard R. Weber (1998). "Optimal robot scheduling for Web search engines". Journal of Scheduling. **1** (1): 15–29. CiteSeerX 10.1.1.36.6087. doi:10.1002/(SICI)1099-1425(199806)1:1<15::AID-JOS3>3.0.CO;2-K.*
31. ^ ***a b*** *Cho, Junghoo; Garcia-Molina, Hector (2003). "Effective page refresh policies for Web crawlers". ACM Transactions on Database Systems. **28** (4): 390–426. doi:10.1145/958942.958945. S2CID 147958.*
32. ^ ***a b*** *Junghoo Cho; Hector Garcia-Molina (2003). "Estimating frequency of change". ACM Transactions on Internet Technology. **3** (3): 256–290. CiteSeerX 10.1.1.59.5877. doi:10.1145/857166.857170. S2CID 9362566.*
33. ^ Ipeirotis, P., Ntoulas, A., Cho, J., Gravano, L. (2005) Modeling and managing content changes in text databases Archived 5 September 2005 at the Wayback Machine. In Proceedings of the 21st IEEE International Conference on Data Engineering, pages 606-617, April 2005, Tokyo.
34. ^ Koster, M. (1995). Robots in the web: threat or treat? ConneXions, 9(4).
35. ^ Koster, M. (1996). A standard for robot exclusion Archived 7 November 2007 at the Wayback Machine.
36. ^ Koster, M. (1993). Guidelines for robots writers Archived 22 April 2005 at the Wayback Machine.
37. ^ Baeza-Yates, R. and Castillo, C. (2002). Balancing volume, quality and freshness in Web crawling. In Soft Computing Systems – Design, Management and Applications, pages 565–572, Santiago, Chile. IOS Press Amsterdam.
38. ^ *Heydon, Allan; Najork, Marc (26 June 1999). "Mercator: A Scalable, Extensible Web Crawler" (PDF). Archived from the original (PDF) on 19 February 2006. Retrieved 22 March 2009.* {{cite journal}}: Cite journal requires |journal= (help)
39. ^ *Dill, S.; Kumar, R.; Mccurley, K. S.; Rajagopalan, S.; Sivakumar, D.; Tomkins, A. (2002). "Self-similarity in the web" (PDF). ACM Transactions on Internet Technology. **2** (3): 205–223. doi:10.1145/572326.572328. S2CID 6416041.*
40. ^ *M. Thelwall; D. Stuart (2006). "Web crawling ethics revisited: Cost, privacy and denial of service". Journal of the American Society for Information Science and Technology. **57** (13): 1771–1779. doi:10.1002/asi.20388.*

41. ^ *Brin, Sergey; Page, Lawrence (1998). "The anatomy of a large-scale hypertextual Web search engine". Computer Networks and ISDN Systems.* ***30*** *(1–7): 107–117. doi: 10.1016/s0169-7552(98)00110-x. S2CID 7587743.*
42. ^ Shkapenyuk, V. and Suel, T. (2002). Design and implementation of a high performance distributed web crawler. In Proceedings of the 18th International Conference on Data Engineering (ICDE), pages 357-368, San Jose, California. IEEE CS Press.
43. ^ Shestakov, Denis (2008). *Search Interfaces on the Web: Querying and Characterizing* Archived 6 July 2014 at the Wayback Machine. TUCS Doctoral Dissertations 104, University of Turku
44. ^ *Michael L Nelson; Herbert Van de Sompel; Xiaoming Liu; Terry L Harrison; Nathan McFarland (24 March 2005). "mod_oai: An Apache Module for Metadata Harvesting": cs/0503069. arXiv:cs/0503069. Bibcode:2005cs.....3069N.* {{cite journal}}: Cite journal requires |journal= (help)
45. ^ *Shestakov, Denis; Bhowmick, Sourav S.; Lim, Ee-Peng (2005). "DEQUE: Querying the Deep Web" (PDF). Data & Knowledge Engineering.* ***52*** *(3): 273–311. doi:10.1016/s0169-023x(04)00107-7.*
46. ^ *"AJAX crawling: Guide for webmasters and developers". Retrieved 17 March 2013.*
47. ^ ITA Labs "ITA Labs Acquisition" Archived 18 March 2014 at the Wayback Machine 20 April 2011 1:28 AM
48. ^ *"About Applebot". Apple Inc. Retrieved 18 October 2021.*
49. ^ *Norton, Quinn (25 January 2007). "Tax takers send in the spiders". Business. Wired. Archived from the original on 22 December 2016. Retrieved 13 October 2017.*
50. ^ *"Xenon web crawling initiative: privacy impact assessment (PIA) summary". Ottawa: Government of Canada. 11 April 2017. Archived from the original on 25 September 2017. Retrieved 13 October 2017.*

Further reading

[edit]

- Cho, Junghoo, "Web Crawling Project", UCLA Computer Science Department.
- A History of Search Engines, from Wiley
- WIVET is a benchmarking project by OWASP, which aims to measure if a web crawler can identify all the hyperlinks in a target website.
- Shestakov, Denis, "Current Challenges in Web Crawling" and "Intelligent Web Crawling", slides for tutorials given at ICWE'13 and WI-IAT'13.

- **v**
- **t**
- **e**

Internet search

Types

- Web search engine (List)
- Metasearch engine
- Multimedia search
- Collaborative search engine
- Cross-language search
- Local search
- Vertical search
- Social search
- Image search
- Audio search
- Video search engine
- Enterprise search
- Semantic search
- Natural language search engine
- Voice search

Tools

- Cross-language information retrieval
- Search by sound
- Search engine marketing
- Search engine optimization
- Evaluation measures
- Search oriented architecture
- Selection-based search
- Document retrieval
- Text mining
- Web crawler
- Multisearch
- Federated search
- Search aggregator
- Index/Web indexing
- Focused crawler
- Spider trap
- Robots exclusion standard
- Distributed web crawling
- Web archiving
- Website mirroring software
- Web query
- Web query classification

Protocols and standards	○ Z39.50 ○ Search/Retrieve Web Service ○ Search/Retrieve via URL ○ OpenSearch ○ Representational State Transfer ○ Wide area information server
See also	○ Search engine ○ Desktop search ○ Online search

○ **v**
○ **t**
○ **e**

Web crawlers

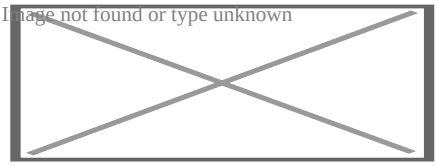
Internet bots designed for Web crawling and Web indexing

Active	○ 80legs ○ bingbot ○ Crawljax ○ Fetcher ○ Googlebot ○ Heritrix ○ HTTrack ○ PowerMapper ○ Wget
Discontinued	○ FAST Crawler ○ msnbot ○ RBSE ○ TkWWW robot ○ Twiceler
Types	○ Distributed web crawler ○ Focused crawler

Authority control databases: National Image not found or type unknown Edit this at Wikidata

About Domain name

This article is about domain names in the Internet. For other uses, see Domain (disambiguation).



An annotated example of a domain name

In the Internet, a **domain name** is a string that identifies a realm of administrative autonomy, authority or control. Domain names are often used to identify services provided through the Internet, such as websites, email services and more. Domain names are used in various networking contexts and for application-specific naming and addressing purposes. In general, a domain name identifies a network domain or an Internet Protocol (IP) resource, such as a personal computer used to access the Internet, or a server computer.

Domain names are formed by the rules and procedures of the Domain Name System (DNS). Any name registered in the DNS is a domain name. Domain names are organized in subordinate levels (*subdomains*) of the DNS root domain, which is nameless. The first-level set of domain names are the *top-level domains* (TLDs), including the *generic top-level domains* (gTLDs), such as the prominent domains com, info, net, edu, and org, and the *country code top-level domains* (ccTLDs). Below these top-level domains in the DNS hierarchy are the second-level and third-level domain names that are typically open for reservation by end-users who wish to connect local area networks to the Internet, create other publicly accessible Internet resources or run websites, such as "wikipedia.org". The registration of a second- or third-level domain name is usually administered by a domain name registrar who sell its services to the public.

A *fully qualified domain name* (FQDN) is a domain name that is completely specified with all labels in the hierarchy of the DNS, having no parts omitted. Traditionally a FQDN ends in a dot (.) to denote the top of the DNS tree.[1] Labels in the Domain Name System are case-insensitive, and may therefore be written in any desired capitalization method, but most commonly domain names are written in lowercase in technical contexts.[2] A *hostname* is a domain name that has at least one associated IP address.

Purpose

[edit]

Domain names serve to identify Internet resources, such as computers, networks, and services, with a text-based label that is easier to memorize than the numerical addresses used in the Internet protocols. A domain name may represent entire collections of such resources or individual instances. Individual Internet host computers use domain names as host identifiers, also called hostnames. The term *hostname* is also used for the leaf labels in the domain name system, usually without further subordinate domain name space. Hostnames appear as a component in Uniform Resource Locators (URLs) for Internet resources such as websites (e.g., en.wikipedia.org).

Domain names are also used as simple identification labels to indicate ownership or control of a resource. Such examples are the realm identifiers used in the Session Initiation Protocol (SIP), the Domain Keys used to verify DNS domains in e-mail systems, and in many other Uniform Resource Identifiers (URIs).

An important function of domain names is to provide easily recognizable and memorizable names to numerically addressed Internet resources. This abstraction allows any resource to be moved to a different physical location in the address topology of the network, globally or locally in an intranet. Such a move usually requires changing the IP address of a resource and the corresponding translation of this IP address to and from its domain name.

Domain names are used to establish a unique identity. Organizations can choose a domain name that corresponds to their name, helping Internet users to reach them easily.

A generic domain is a name that defines a general category, rather than a specific or personal instance, for example, the name of an industry, rather than a company name. Some examples of generic names are *books.com*, *music.com*, and *travel.info*. Companies have created brands based on generic names, and such generic domain names may be valuable.[3]

Domain names are often simply referred to as *domains* and domain name registrants are frequently referred to as *domain owners*, although domain name registration with a registrar does not confer any legal ownership of the domain name, only an exclusive right of use for a particular duration of time. The use of domain names in commerce may subject them to trademark law.

History

[edit]
Main article: List of the oldest currently registered Internet domain names

The practice of using a simple memorable abstraction of a host's numerical address on a computer network dates back to the ARPANET era, before the advent of today's commercial Internet. In the early network, each computer on the network retrieved the hosts file (*host.txt*) from a computer at SRI (now SRI International),[4][5] which mapped computer hostnames to numerical addresses. The rapid growth of the network made it impossible to maintain a centrally organized hostname registry and in 1983 the Domain Name System was introduced on the ARPANET and published by the Internet Engineering Task Force as RFC 882 and RFC 883.

The following table shows the first five .com domains with the dates of their registration:[6]

Domain name	Registration date
symbolics.com	15 March 1985
bbn.com	24 April 1985
think.com	24 May 1985
mcc.com	11 July 1985
dec.com	30 September 1985

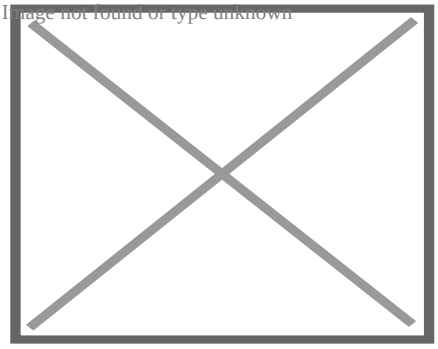
and the first five .edu domains:[7]

Domain name	Registration date
berkeley.edu	24 April 1985
cmu.edu	24 April 1985
purdue.edu	24 April 1985
rice.edu	24 April 1985
ucla.edu	24 April 1985

Domain name space

[edit]

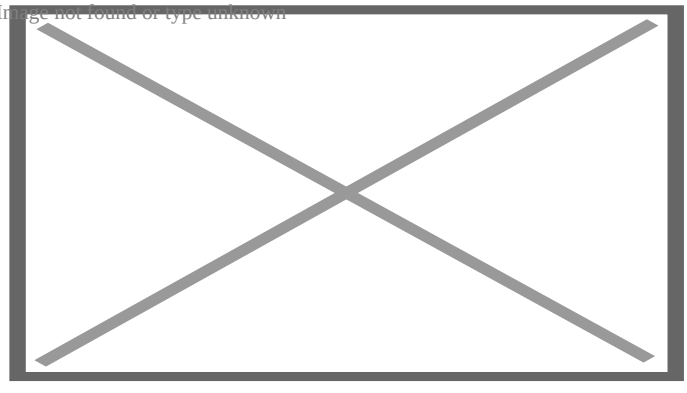
Image not found or type unknown



The hierarchical domain name system, organized into zones, each served by domain name servers

Today, the Internet Corporation for Assigned Names and Numbers (ICANN) manages the top-level development and architecture of the Internet domain name space. It authorizes domain name registrars, through which domain names may be registered and reassigned.

Image not found or type unknown



The hierarchy of labels in a fully qualified domain name

The domain name space consists of a tree of domain names. Each node in the tree holds information associated with the domain name. The tree sub-divides into *zones* beginning at the DNS root zone.

Domain name syntax

[edit]

A domain name consists of one or more parts, technically called *labels*, that are conventionally concatenated, and delimited by dots, such as *example.com*.

- The right-most label conveys the top-level domain; for example, the domain name *www.example.com* belongs to the top-level domain *com*.
- The hierarchy of domains descends from the right to the left label in the name; each label to the left specifies a subdivision, or subdomain of the domain to the right. For example: the label *example* specifies a node *example.com* as a subdomain of the *com* domain, and *www* is a label to create *www.example.com*, a subdomain of *example.com*. Each label may contain from 1 to 63 octets. The empty label is reserved for the root node and when fully qualified is expressed as the empty label terminated by a dot. The full domain name may not exceed a total length of 253 ASCII characters in its textual representation.[8]
- A hostname is a domain name that has at least one associated IP address. For example, the domain names *www.example.com* and *example.com* are also hostnames, whereas the *com* domain is not. However, other top-level domains, particularly country code top-level domains, may indeed have an IP address, and if so, they are also hostnames.
- Hostnames impose restrictions on the characters allowed in the corresponding domain name. A valid hostname is also a valid domain name, but a valid domain name may not necessarily be valid as a hostname.

Top-level domains

[edit]

When the Domain Name System was devised in the 1980s, the domain name space was divided into two main groups of domains.[9] The country code top-level domains (ccTLD) were primarily based on the two-character territory codes of ISO-3166 country abbreviations. In addition, a group of seven generic top-level domains (gTLD) was implemented which represented a set of categories of names and multi-organizations.[10] These were the domains gov, edu, com, mil, org, net, and int. These two types of top-level domains (TLDs) are the highest level of domain names of the Internet. Top-level domains form the DNS root zone of the hierarchical Domain Name System. Every domain name ends with a top-level domain label.

During the growth of the Internet, it became desirable to create additional generic top-level domains. As of October 2009, 21 generic top-level domains and 250 two-letter country-code top-

level domains existed.[11] In addition, the ARPA domain serves technical purposes in the infrastructure of the Domain Name System.

During the 32nd International Public ICANN Meeting in Paris in 2008,[12] ICANN started a new process of TLD naming policy to take a "significant step forward on the introduction of new generic top-level domains." This program envisions the availability of many new or already proposed domains, as well as a new application and implementation process.[13] Observers believed that the new rules could result in hundreds of new top-level domains to be registered.[14] In 2012, the program commenced, and received 1930 applications.[15] By 2016, the milestone of 1000 live gTLD was reached.

The Internet Assigned Numbers Authority (IANA) maintains an annotated list of top-level domains in the DNS root zone database.[16]

For special purposes, such as network testing, documentation, and other applications, IANA also reserves a set of special-use domain names.[17] This list contains domain names such as example, local, localhost, and test. Other top-level domain names containing trade marks are registered for corporate use. Cases include brands such as BMW, Google, and Canon.[18]

Second-level and lower level domains

[edit]

Below the top-level domains in the domain name hierarchy are the second-level domain (SLD) names. These are the names directly to the left of .com, .net, and the other top-level domains. As an example, in the domain *example.co.uk*, *co* is the second-level domain.

Next are third-level domains, which are written immediately to the left of a second-level domain. There can be fourth- and fifth-level domains, and so on, with virtually no limitation. Each label is separated by a full stop (dot). An example of an operational domain name with four levels of domain labels is *sos.state.oh.us*. 'sos' is said to be a sub-domain of 'state.oh.us', and 'state' a sub-domain of 'oh.us', etc. In general, subdomains are domains subordinate to their parent domain. An example of very deep levels of subdomain ordering are the IPv6 reverse resolution DNS zones, e.g., 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa, which is the reverse DNS resolution domain name for the IP address of a loopback interface, or the localhost name.

Second-level (or lower-level, depending on the established parent hierarchy) domain names are often created based on the name of a company (e.g., *bbc*.co.uk), product or service (e.g. *hotmail*.com). Below these levels, the next domain name component has been used to designate a particular host server. Therefore, *ftp.example.com* might be an FTP server, *www.example.com* would be a World Wide Web server, and *mail.example.com* could be an email server, each intended to perform only the implied function. Modern technology allows multiple physical servers with either different (cf. load balancing) or even identical addresses (cf. anycast) to serve a single hostname or domain name, or multiple domain names to be served by a single computer. The latter is very popular in Web hosting service centers, where service providers host the websites of many organizations on just a few servers.

The hierarchical DNS labels or components of domain names are separated in a fully qualified name by the full stop (dot, .).

Internationalized domain names

[edit]
Main article: Internationalized domain name

The character set allowed in the Domain Name System is based on ASCII and does not allow the representation of names and words of many languages in their native scripts or alphabets. ICANN approved the Internationalized domain name (IDNA) system, which maps Unicode strings used in application user interfaces into the valid DNS character set by an encoding called Punycode. For example, københavn.eu is mapped to xn--kbenhavn-54a.eu. Many registries have adopted IDNA.

Domain name registration

[edit]

History

[edit]

The first commercial Internet domain name, in the TLD *com*, was registered on 15 March 1985 in the name symbolics.com by Symbolics Inc., a computer systems firm in Cambridge, Massachusetts.

By 1992, fewer than 15,000 *com* domains had been registered.

In the first quarter of 2015, 294 million domain names had been registered.[19] A large fraction of them are in the *com* TLD, which as of December 21, 2014, had 115.6 million domain names,[20] including 11.9 million online business and e-commerce sites, 4.3 million entertainment sites, 3.1 million finance related sites, and 1.8 million sports sites.[21] As of July 15, 2012, the *com* TLD had more registrations than all of the ccTLDs combined.[22]

As of December 31, 2023, 359.8 million domain names had been registered.[23]

Administration

[edit]

The right to use a domain name is delegated by domain name registrars, which are accredited by the Internet Corporation for Assigned Names and Numbers (ICANN), the organization charged with overseeing the name and number systems of the Internet. In addition to ICANN, each top-level domain (TLD) is maintained and serviced technically by an administrative organization operating a registry. A registry is responsible for maintaining the database of names registered within the TLD it administers. The registry receives registration information from each domain

name registrar authorized to assign names in the corresponding TLD and publishes the information using a special service, the WHOIS protocol.

Registries and registrars usually charge an annual fee for the service of delegating a domain name to a user and providing a default set of name servers. Often, this transaction is termed a sale or lease of the domain name, and the registrant may sometimes be called an "owner", but no such legal relationship is actually associated with the transaction, only the exclusive right to use the domain name. More correctly, authorized users are known as "registrants" or as "domain holders".

ICANN publishes the complete list of TLD registries and domain name registrars. Registrant information associated with domain names is maintained in an online database accessible with the WHOIS protocol. For most of the 250 country code top-level domains (ccTLDs), the domain registries maintain the WHOIS (Registrant, name servers, expiration dates, etc.) information.

Some domain name registries, often called *network information centers* (NIC), also function as registrars to end-users. The major generic top-level domain registries, such as for the *com*, *net*, *org*, *info* domains and others, use a registry-registrar model consisting of hundreds of domain name registrars (see lists at ICANN[24] or VeriSign).[25] In this method of management, the registry only manages the domain name database and the relationship with the registrars. The *registrants* (users of a domain name) are customers of the registrar, in some cases through additional layers of resellers.

There are also a few other alternative DNS root providers that try to compete or complement ICANN's role of domain name administration, however, most of them failed to receive wide recognition, and thus domain names offered by those alternative roots cannot be used universally on most other internet-connecting machines without additional dedicated configurations.

Technical requirements and process

[edit]

In the process of registering a domain name and maintaining authority over the new name space created, registrars use several key pieces of information connected with a domain:

- *Administrative contact*. A registrant usually designates an administrative contact to manage the domain name. The administrative contact usually has the highest level of control over a domain. Management functions delegated to the administrative contacts may include management of all business information, such as name of record, postal address, and contact information of the official registrant of the domain and the obligation to conform to the requirements of the domain registry in order to retain the right to use a domain name. Furthermore, the administrative contact installs additional contact information for technical and billing functions.
- *Technical contact*. The technical contact manages the name servers of a domain name. The functions of a technical contact include assuring conformance of the configurations of the domain name with the requirements of the domain registry, maintaining the domain zone records, and providing continuous functionality of the name servers (that leads to the

accessibility of the domain name).
- *Billing contact*. The party responsible for receiving billing invoices from the domain name registrar and paying applicable fees.
- *Name servers*. Most registrars provide two or more name servers as part of the registration service. However, a registrant may specify its own authoritative name servers to host a domain's resource records. The registrar's policies govern the number of servers and the type of server information required. Some providers require a hostname and the corresponding IP address or just the hostname, which must be resolvable either in the new domain, or exist elsewhere. Based on traditional requirements (RFC 1034), typically a minimum of two servers is required.

A domain name consists of one or more labels, each of which is formed from the set of ASCII letters, digits, and hyphens (a–z, A–Z, 0–9, -), but not starting or ending with a hyphen. The labels are case-insensitive; for example, 'label' is equivalent to 'Label' or 'LABEL'. In the textual representation of a domain name, the labels are separated by a full stop (period).

Business models

[edit]

Domain names are often seen in analogy to real estate in that domain names are foundations on which a website can be built, and the highest *quality* domain names, like sought-after real estate, tend to carry significant value, usually due to their online brand-building potential, use in advertising, search engine optimization, and many other criteria.

A few companies have offered low-cost, below-cost or even free domain registration with a variety of models adopted to recoup the costs to the provider. These usually require that domains be hosted on their website within a framework or portal that includes advertising wrapped around the domain holder's content, revenue from which allows the provider to recoup the costs. Domain registrations were free of charge when the DNS was new. A domain holder may provide an infinite number of subdomains in their domain. For example, the owner of *example.org* could provide subdomains such as *foo.example.org* and *foo.bar.example.org* to interested parties.

Many desirable domain names are already assigned and users must search for other acceptable names, using Web-based search features, or WHOIS and dig operating system tools. Many registrars have implemented **domain name suggestion** tools which search domain name databases and suggest available alternative domain names related to keywords provided by the user.

Resale of domain names

[edit]
Main article: List of most expensive domain names

The business of resale of registered domain names is known as the domain aftermarket. Various factors influence the perceived value or market value of a domain name. Most of the high-prize

domain sales are carried out privately.[26] Also, it is called confidential domain acquiring or anonymous domain acquiring.[27]

Domain name confusion

[edit]

Intercapping is often used to emphasize the meaning of a domain name, because DNS names are not case-sensitive. Some names may be misinterpreted in certain uses of capitalization. For example: *Who Represents*, a database of artists and agents, chose *whorepresents.com*,[28] which can be misread. In such situations, the proper meaning may be clarified by placement of hyphens when registering a domain name. For instance, Experts Exchange, a programmers' discussion site, used *expertsexchange.com*, but changed its domain name to *experts-exchange.com*.[29]

Uses in website hosting

[edit]

The domain name is a component of a uniform resource locator (URL) used to access websites, for example:

- URL: http://www.example.net/index.html
- Top-level domain: net
- Second-level domain: example
- Hostname: www

A domain name may point to multiple IP addresses to provide server redundancy for the services offered, a feature that is used to manage the traffic of large, popular websites.

Web hosting services, on the other hand, run servers that are typically assigned only one or a few addresses while serving websites for many domains, a technique referred to as virtual web hosting . Such IP address overloading requires that each request identifies the domain name being referenced, for instance by using the HTTP request header field *Host:*, or Server Name Indication.

Abuse and regulation

[edit]

Critics often claim abuse of administrative power over domain names. Particularly noteworthy was the VeriSign Site Finder system which redirected all unregistered .com and .net domains to a VeriSign webpage. For example, at a public meeting with VeriSign to air technical concerns about Site Finder,[30] numerous people, active in the IETF and other technical bodies, explained how they were surprised by VeriSign's changing the fundamental behavior of a major component of Internet infrastructure, not having obtained the customary consensus. Site Finder, at first, assumed every Internet query was for a website, and it monetized queries for incorrect domain names, taking the user to VeriSign's search site. Other applications, such as many implementations of

email, treat a lack of response to a domain name query as an indication that the domain does not exist, and that the message can be treated as undeliverable. The original VeriSign implementation broke this assumption for mail, because it would always resolve an erroneous domain name to that of Site Finder. While VeriSign later changed Site Finder's behaviour with regard to email, there was still widespread protest about VeriSign's action being more in its financial interest than in the interest of the Internet infrastructure component for which VeriSign was the steward.

Despite widespread criticism, VeriSign only reluctantly removed it after the Internet Corporation for Assigned Names and Numbers (ICANN) threatened to revoke its contract to administer the root name servers. ICANN published the extensive set of letters exchanged, committee reports, and ICANN decisions.[31]

There is also significant disquiet regarding the United States Government's political influence over ICANN. This was a significant issue in the attempt to create a .xxx top-level domain and sparked greater interest in alternative DNS roots that would be beyond the control of any single country.[32]

Additionally, there are numerous accusations of domain name front running, whereby registrars, when given whois queries, automatically register the domain name for themselves. Network Solutions has been accused of this.[33]

Truth in Domain Names Act

[edit]

In the United States, the Truth in Domain Names Act of 2003, in combination with the PROTECT Act of 2003, forbids the use of a misleading domain name with the intention of attracting Internet users into visiting Internet pornography sites.

The Truth in Domain Names Act follows the more general Anticybersquatting Consumer Protection Act passed in 1999 aimed at preventing typosquatting and deceptive use of names and trademarks in domain names.

Seizures

[edit]

- Seizure notices absolutepoker.com
- Image not found or type unknown

absolutepoker.com

channelsurfing.net

In the early 21st century, the US Department of Justice (DOJ) pursued the seizure of domain names, based on the legal theory that domain names constitute property used to engage in criminal activity, and thus are subject to forfeiture. For example, in the seizure of the domain name of a gambling website, the DOJ referenced 18 U.S.C. § 981 and 18 U.S.C. § 1955(d).[34][1] In 2013 the US government seized Liberty Reserve, citing 18 U.S.C. § 982(a)(1).[35]

- Image not found or type unknown

channelsurfing.net
libertyreserve.com

The U.S. Congress passed the Combating Online Infringement and Counterfeits Act in 2010. Consumer Electronics Association vice president Michael Petricone was worried that seizure was a *blunt instrument* that could harm legitimate businesses.[36][37] After a joint operation on February 15, 2011, the DOJ and the Department of Homeland Security claimed to have seized ten domains of websites involved in advertising and distributing child pornography, but also mistakenly seized the domain name of a large DNS provider, temporarily replacing 84,000 websites with seizure notices.[38]

- Image not found or type unknown

libertyreserve.com

In the United Kingdom, the Police Intellectual Property Crime Unit (PIPCU) has been attempting to seize domain names from registrars without court orders.[39]

Suspensions

[edit]

PIPCU and other UK law enforcement organisations make domain suspension requests to Nominet which they process on the basis of breach of terms and conditions. Around 16,000 domains are suspended annually, and about 80% of the requests originate from PIPCU.[40]

Property rights

[edit]

Because of the economic value it represents, the European Court of Human Rights has ruled that the exclusive right to a domain name is protected as property under article 1 of Protocol 1 to the European Convention on Human Rights.[41]

IDN variants

[edit]

ICANN Business Constituency (BC) has spent decades trying to make IDN variants work at the second level, and in the last several years at the top level. Domain name variants are domain names recognized in different character encodings, like a single domain presented in traditional

Chinese and simplified Chinese. It is an Internationalization and localization problem. Under Domain Name Variants, the different encodings of the domain name (in simplified and traditional Chinese) would resolve to the same host.[42][43]

According to John Levine, an expert on Internet related topics, "Unfortunately, variants don't work. The problem isn't putting them in the DNS, it's that once they're in the DNS, they don't work anywhere else."[42]

Fictitious domain name

[edit]

A *fictitious domain name* is a domain name used in a work of fiction or popular culture to refer to a domain that does not actually exist, often with invalid or unofficial top-level domains such as ".web ", a usage exactly analogous to the dummy 555 telephone number prefix used in film and other media. The canonical fictitious domain name is "example.com", specifically set aside by IANA in RFC 2606 for such use, along with the *.example* TLD.

Domain names used in works of fiction have often been registered in the DNS, either by their creators or by cybersquatters attempting to profit from it. This phenomenon prompted NBC to purchase the domain name Hornymanatee.com after talk-show host Conan O'Brien spoke the name while ad-libbing on his show. O'Brien subsequently created a website based on the concept and used it as a running gag on the show.[44] Companies whose works have used fictitious domain names have also employed firms such as MarkMonitor to park fictional domain names in order to prevent misuse by third parties.[45]

Misspelled domain names

[edit]

This section **does not cite any sources**. Please help improve this section by adding citations to reliable sources. Unsourced material may be challenged and removed. *(December 2022) (Learn how and when to remove this message)*

Misspelled domain names, also known as typosquatting or URL hijacking, are domain names that are intentionally or unintentionally misspelled versions of popular or well-known domain names. The goal of misspelled domain names is to capitalize on internet users who accidentally type in a misspelled domain name, and are then redirected to a different website.

Misspelled domain names are often used for malicious purposes, such as phishing scams or distributing malware. In some cases, the owners of misspelled domain names may also attempt to sell the domain names to the owners of the legitimate domain names, or to individuals or organizations who are interested in capitalizing on the traffic generated by internet users who accidentally type in the misspelled domain names.

To avoid being caught by a misspelled domain name, internet users should be careful to type in domain names correctly, and should avoid clicking on links that appear suspicious or unfamiliar.

Additionally, individuals and organizations who own popular or well-known domain names should consider registering common misspellings of their domain names in order to prevent others from using them for malicious purposes.

Domain name spoofing

[edit]

The term **Domain name spoofing** (or simply though less accurately, **Domain spoofing**) is used generically to describe one or more of a class of phishing attacks that depend on falsifying or misrepresenting an internet domain name.[46][47] These are designed to persuade unsuspecting users into visiting a web site other than that intended, or opening an email that is not in reality from the address shown (or apparently shown).[48] Although website and email spoofing attacks are more widely known, any service that relies on domain name resolution may be compromised.

Types

[edit]

There are a number of better-known types of domain spoofing:

- Typosquatting, also called "URL hijacking", a "sting site", or a "fake URL", is a form of cybersquatting, and possibly brandjacking which relies on mistakes such as typos made by Internet users when inputting a website address into a web browser or composing an email address. Should a user accidentally enter an incorrect domain name, they may be led to any URL (including an alternative website owned by a cybersquatter).[49]

  The typosquatter's URL will usually be one of five kinds, all *similar to* the victim site address:
  - A common misspelling, or foreign language spelling, of the intended site
  - A misspelling based on a typographical error
  - A plural of a singular domain name
  - A different top-level domain: (i.e. .com instead of .org)
  - An abuse of the Country Code Top-Level Domain (ccTLD) (.cm, .co, or .om instead of .com)

- IDN homograph attack. This type of attack depends on registering a domain name that is similar to the 'target' domain, differing from it only because its spelling includes one or more characters that come from a different alphabet but look the same to the naked eye. For example, the Cyrillic, Latin, and Greek alphabets each have their own letter A, each of which has its own binary code point. Turkish has a dotless letter i (ÃƒÆ'Ã†â€™Ãƒâ€ Ã¢â‚¬â„¢ÃƒÆ'Ã¢â‚¬Â ÃƒÂ¢Ã¢â€šÂ¬Ã¢â€žÂ¢ÃƒÆ'Ã†â€™ÃƒÂ¢Ã¢â€šÂ¬Ã‚Ã‚) that may not be perceived as different from the ASCII letter i. Most web browsers warn of 'mixed alphabet' domain names,[50][51][52][53] Other services, such as email applications, may not provide the same protection. Reputable top level domain and country code domain registrars will not accept applications to register a deceptive name but this policy cannot be presumed to be infallible.

- DNS spoofing – Cyberattack using corrupt DNS data
- Website spoofing – Creating a website, as a hoax, with the intention of misleading readers
- Email spoofing – Creating email spam or phishing messages with a forged sender identity or address

Risk mitigation

[edit]

- Domain Name System Security Extensions – Suite of IETF specifications
- Sender Policy Framework – Simple email-validation system designed to detect email spoofing
- DMARC – System to prevent email fraud ("Domain-based Message Authentication, Reporting and Conformance")
- DomainKeys Identified Mail – Email authentication method designed to detect email spoofing
- Public key certificate – Electronic document used to prove the ownership of a public key (SSL certificate)

Legitimate technologies that may be subverted

[edit]

- URL redirection – Technique for making a Web page available under more than one URL address
- Domain fronting – Technique for Internet censorship circumvention

See also

[edit]

- Domain hack
- Domain hijacking
- Domain name registrar
- Domain name speculation
- Domain name warehousing
- Domain registration
- Domain tasting
- Geodomain
- List of Internet top-level domains
- Reverse domain hijacking
- Reverse domain name notation

References

[edit]

1. ^ *Stevens, W. Richard (1994). TCP/IP Illustrated, Volume 1: The Protocols. Vol. 1 (1 ed.). Addison-Wesley. ISBN 9780201633467.*
2. ^ *Arends, R.; Austein, R.; Larson, M.; Massey, D.; Rose, S. (2005). RFC 4034 – Resource Records for the DNS Security Extensions (Technical report). IEFT. doi:10.17487/RFC4034. Archived from the original on 2018-09-20. Retrieved 2015-07-05.*
3. ^ *Low, Jerry. "Why are generic domains so expensive?". TheRealJerryLow.com. Archived from the original on 20 March 2019. Retrieved 27 September 2018.*
4. ^ RFC 3467, *Role of the Domain Name System (DNS)*, J.C. Klensin, J. Klensin (February 2003)
5. ^ *Cricket Liu, Paul Albitz (2006). DNS and BIND (5th ed.). O'Reilly. p. 3. Archived from the original on 2011-09-05. Retrieved 2011-10-22.*
6. ^ *"The first ever 20 domain names registered". ComputerWeekly.com. Archived from the original on 2020-08-08. Retrieved 2020-07-30.*
7. ^ *Rooksby, Jacob H. (2015). "Defining Domain: Higher Education's Battles for Cyberspace". Brooklyn Law Review. **80** (3): 857–942. Archived from the original on 2018-11-07. Retrieved 2015-10-27.* at p. 869
8. ^ *Mockapetris, P. (November 1987). "Domain names - Implementation and specification (RFC 1035)". IETF Datatracker. Retrieved January 21, 2024.*
9. ^ *"Introduction to Top-Level Domains (gTLDs)". Internet Corporation for Assigned Names and Numbers (ICANN). Archived from the original on 2009-06-15. Retrieved 2009-06-26.*
10. ^ RFC 920, *Domain Requirements*, J. Postel, J. Reynolds, The Internet Society (October 1984)
11. ^ "New gTLD Program" Archived 2011-11-25 at the Wayback Machine, ICANN, October 2009
12. ^ *"32nd International Public ICANN Meeting". ICANN. 2008-06-22. Archived from the original on 2009-03-08. Retrieved 2009-06-26.*
13. ^ *"New gTLS Program". ICANN. Archived from the original on 2011-09-10. Retrieved 2009-06-15.*
14. ^ ICANN Board Approves Sweeping Overhaul of Top-level Domains Archived 2009-06-26 at the Wayback Machine, CircleID, 26 June 2008.
15. ^ *"About the Program - ICANN New gTLDs". ICANN. Archived from the original on 2016-11-03. Retrieved 2016-11-09.*
16. ^ *"Root Zone Database". IANA. Archived from the original on 2019-05-04. Retrieved 2020-11-01.*
17. ^ *Cheshire, S.; Krochmal, M. (February 2013). "RFC6761 - Special-Use Domain Names". Internet Engineering Task Force. doi:10.17487/RFC6761. Archived from the original on 13 November 2020. Retrieved 3 May 2015.*
18. ^ *"Executive Summary - dot brand observatory". observatory.domains. Archived from the original on 2016-11-10. Retrieved 2016-11-09.*
19. ^ Internet Grows to 294 Million Domain Names in the First Quarter of 2015 Archived 2017-12-20 at the Wayback Machine, Jun 30, 2015.
20. ^ *"Thirty years of .COM domains - and the numbers are up". Geekzone. Mar 13, 2015. Archived from the original on April 7, 2016. Retrieved Mar 25, 2016.*
21. ^ Evangelista, Benny. 2010. "25 years of .com names." San Francisco Chronicle. March 15, p. 1

22. ^ *"Domain domination: The com TLD larger than all ccTLDs combined". Royal.pingdom.com. Archived from the original on 2012-07-23. Retrieved 2012-07-25.*
23. ^ *"DNIB Quarterly Report Q4 2023". Domain Name Industry Brief (DNIB). Retrieved 16 February 2024.*
24. ^ *"ICANN-Accredited Registrars". ICANN. Archived from the original on 2019-05-19. Retrieved 2012-09-13.*
25. ^ *"Choose A Top Domain Registrar Of Your Choice Using Our Search Tool". Verisign. Archived from the original on 2015-09-04. Retrieved 2015-08-10.*
26. ^ *Arif, Sengoren (1 October 2024). "Confidentially domain acquiring".*
27. ^ *"Anonymous Domain Ownership". Conference: 2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). 1 October 2024.*
28. ^ *Courtney, Curzi (14 October 2014). "WhoRepresents helps brands connect with celebrity influencers". DM News. Archived from the original on 8 July 2019. Retrieved 8 July 2019.*
29. ^ *Ki, Mae Heussner (2 June 2010). "'Slurls': Most Outrageous Website URLs". ABC News. Archived from the original on 31 May 2019. Retrieved 8 July 2019.*
30. ^ *McCullagh, Declan (2003-10-03). "VeriSign fends off critics at ICANN confab". CNET News. Archived from the original on January 4, 2013. Retrieved 2007-09-22.*
31. ^ *"Verisign's Wildcard Service Deployment". ICANN. Archived from the original on 2008-12-02. Retrieved 2007-09-22.*
32. ^ *Mueller, M (March 2004). Ruling the Root. MIT Press. ISBN 0-262-63298-5.*
33. ^ Slashdot.org Archived 2010-02-17 at the Wayback Machine, NSI Registers Every Domain Checked
34. ^ *FBI / DOJ (15 April 2011). "Warning". Archived from the original on 2011-04-14. Retrieved 2011-04-15.*
35. ^ *Dia, Miaz (4 February 2010). "website laten maken". Kmowebdiensten. Archived from the original on December 20, 2016. Retrieved 8 December 2016.*
36. ^ *Gabriel, Jeffrey (18 June 2020). "Past Congressional Attempts to Combat Online Copyright Infringement". Saw. Archived from the original on 2020-06-20. Retrieved 2020-06-19.*
37. ^ *Jerome, Sarah (6 April 2011). "Tech industry wary of domain name seizures". The Hill. Archived from the original on 2011-04-10. Retrieved 2011-04-15.*
38. ^ *"U.S. Government Shuts Down 84,000 Websites, 'By Mistake'". Archived from the original on 2018-12-25. Retrieved 2012-12-16.*
39. ^ *Jeftovic, Mark (8 October 2013). "Whatever Happened to "Due Process" ?". Archived from the original on 5 December 2014. Retrieved 27 November 2014.*
40. ^ Tackling online criminal activity Archived 2017-12-16 at the Wayback Machine, 1 November 2016 – 31 October 2017, Nominet
41. ^ ECHR 18 September 2007, no. 25379/04, 21688/05, 21722/05, 21770/05, *Paeffgen v Germany*.
42. ^ ***a b*** *Levine, John R. (April 21, 2019). "Domain Name Variants Still Won't Work". Archived from the original on July 29, 2020. Retrieved May 23, 2020.*
43. ^ *"Comment on ICANN Recommendations for Managing IDN Variant Top-Level Domains" (PDF). ICANN. April 21, 2019. Archived (PDF) from the original on 2022-10-09. Retrieved May 23, 2020.*
44. ^ "So This Manatee Walks Into the Internet Archived 2017-01-23 at the Wayback Machine", *The New York Times*, December 12, 2006. Retrieved April 12, 2008.

45. ^ *Allemann, Andrew (2019-11-05). "Part of MarkMonitor sold to OpSec Security". Domain Name Wire | Domain Name News. Retrieved 2024-11-26.*
46. ^ *"Canadian banks hit by two-year domain name spoofing scam". Finextra. 9 January 2020. Archived from the original on 6 November 2021. Retrieved 27 August 2021.*
47. ^ *"Domain spoofing". Barracuda Networks. Archived from the original on 2021-11-04. Retrieved 2021-08-27.*
48. ^ *Tara Seals (August 6, 2019). "Mass Spoofing Campaign Abuses Walmart Brand". threatpost. Archived from the original on November 6, 2021. Retrieved August 27, 2021.*
49. ^ *"Example Screenshots of Strider URL Tracer With Typo-Patrol". Microsoft Research. Archived from the original on 21 December 2008.*
50. ^ *"Internationalized Domain Names (IDN) in Google Chrome". chromium.googlesource.com. Archived from the original on 2020-11-01. Retrieved 2020-08-26.*
51. ^ *"Upcoming update with IDN homograph phishing fix - Blog". Opera Security. 2017-04-21. Archived from the original on 2020-08-08. Retrieved 2020-08-26.*
52. ^ *"About Safari International Domain Name support". Archived from the original on 2014-06-17. Retrieved 2017-04-29.*
53. ^ *"IDN Display Algorithm". Mozilla. Archived from the original on 2016-01-31. Retrieved 2016-01-31.*

External links

[edit]

Image not found or type unknown

Look up ***homograph*** in Wiktionary, the free dictionary.

Image not found or type unknown

Wikimedia Commons has media related to ***Domain name space***.

- (domain bias in web search) a research by Microsoft
- Top Level Domain Bias in Search Engine Indexing and Rankings
- Icann New gTLD Program Factsheet - October 2009 (PDF)
- IANA Two letter Country Code TLD
- ICANN - Internet Corporation for Assigned Names and Numbers
- Internic.net, public information regarding Internet domain name registration services
- Internet Domain Names: Background and Policy Issues Congressional Research Service
- RFC 1034, Domain Names — Concepts and Facilities, an Internet Protocol Standard
- RFC 1035, Domain Names — Implementation and Specification, an Internet Protocol Standard
- UDRP, Uniform Domain-Name Dispute-Resolution Policy
- Special use domain names

- **v**
- **t**

- **e**

Website management

Concepts

Web hosting
- Clustered
- Peer-to-peer
- Self-hosting
- Virtual

Web analytics
- Click analytics
- Mobile web analytics
- Web tracking
  - Click tracking

- Overselling
- Web document
- Web content
- Web content lifecycle
- Web server
- Web cache
- Webmaster
- Website governance

Web hosting control panels (comparison)
- AlternC
- cPanel
- DirectAdmin
- Domain Technologie Control
- Froxlor
- i-MSCP
- InterWorx
- ISPConfig
- Ispmanager
- Kloxo
- Plesk
- Usermin
- Webmin

Top-level domain registries

- AFNIC
- auDA
- DNS Belgium
- CentralNic
- CIRA
- CNNIC
- CZ.NIC
- DENIC
- EURid
- Freenom
- GoDaddy
- Google Domains
- Identity Digital
- IPM
- JPRS
- KISA
- NIC México
- Nominet
- PIR
- Tucows
- Verisign

Domain name managers and registrars	○ Bluehost ○ Domainz ○ DreamHost ○ Dynadot ○ Enom ○ Epik ○ Gandi ○ GlowHost ○ GMO Internet ○ GoDaddy ○ Google Domains ○ Hover ○ Infomaniak ○ Jimdo ○ Name.com ○ Namecheap ○ Hostinger ○ NameSilo ○ NearlyFreeSpeech ○ Network Solutions ○ OVH ○ Register.com ○ Squarespace ○ Tucows ○ UK2 ○ Webcentral ○ Web.com ○ Wix.com
Web content management system	○ Document management system ○ Wiki software ○ Blog software
Authority control databases: National Image not found or type unknown **Edit this at Wikidata**	○ Germany ○ United States ○ France ○ BnF data ○ Japan ○ Israel

About Web syndication

Web syndication is making content available from one website to other sites. Most commonly, websites are made available to provide either summaries or full renditions of a website's recently added content. The term may also describe other kinds of content licensing for reuse.

Motivation

[edit]

For the subscribing sites, syndication is an effective way of adding greater depth and immediacy of information to their pages, making them more attractive to users. For the provider site, syndication increases exposure. This generates new traffic for the provider site—making syndication an easy and relatively cheap, or even free, form of advertisement.

Content syndication has become an effective strategy for link building, as search engine optimization has become an increasingly important topic among website owners and online marketers. Links embedded within the syndicated content are typically optimized around anchor terms that will point an optimized[*clarification needed*] link back to the website that the content author is trying to promote. These links tell the algorithms of the search engines that the website being linked to is an authority for the keyword that is being used as the anchor text. However the rollout of Google Panda's algorithm may not reflect this authority in its SERP rankings based on quality scores generated by the sites linking to the authority.

The prevalence of web syndication is also of note to online marketers, since web surfers are becoming increasingly wary of providing personal information for marketing materials (such as signing up for a newsletter) and expect the ability to subscribe to a feed instead. Although the format could be anything transported over HTTP, such as HTML or JavaScript, it is more commonly XML. Web syndication formats include RSS, Atom,[1] and JSON Feed.

History

[edit]
Main article: History of web syndication technology

Syndication first arose in earlier media such as print, radio, and television, allowing content creators to reach a wider audience. In the case of radio, the United States Federal government proposed a syndicate in 1924 so that the country's executives could quickly and efficiently reach the entire population.[2] In the case of television, it is often said that "Syndication is where the real money is."[3] Additionally, syndication accounts for the bulk of TV programming.[4]

One predecessor of web syndication is the Meta Content Framework (MCF), developed in 1996 by Ramanathan V. Guha and others in Apple Computer's Advanced Technology Group.[5]

Today, millions of online publishers, including newspapers, commercial websites, and blogs, distribute their news headlines, product offers, and blog postings in the news feed.

As a commercial model

[edit]

Conventional syndication businesses such as Reuters and Associated Press thrive on the internet by offering their content to media partners on a subscription basis,[6] using business models established in earlier media forms.

Commercial web syndication can be categorized in three ways:

- by *business models*
- by *types of content*
- by *methods for selecting distribution partners*

Commercial web syndication involves partnerships between content producers and distribution outlets. There are different structures of partnership agreements. One such structure is licensing content, in which distribution partners pay a fee to the content creators for the right to publish the content. Another structure is ad-supported content, in which publishers share revenues derived from advertising on syndicated content with that content's producer. A third structure is free, or barter syndication, in which no currency changes hands between publishers and content producers. This requires the content producers to generate revenue from another source, such as embedded advertising or subscriptions. Alternatively, they could distribute content without remuneration. Typically, those who create and distribute content free are promotional entities, vanity publishers, or government entities.

Types of content syndicated include RSS or Atom Feeds and full content. With RSS feeds, headlines, summaries, and sometimes a modified version of the original full content is displayed on users' feed readers. With full content, the entire content—which might be text, audio, video, applications/widgets, or user-generated content—appears unaltered on the publisher's site.

There are two methods for selecting distribution partners. The content creator can hand-pick syndication partners based on specific criteria, such as the size or quality of their audiences. Alternatively, the content creator can allow publisher sites or users to opt into carrying the content through an automated system. Some of these automated "content marketplace" systems involve careful screening of potential publishers by the content creator to ensure that the material does not end up in an inappropriate environment.

Just as syndication is a source of profit for TV producers and radio producers, it also functions to maximize profit for Internet content producers. As the Internet has increased in size[7] it has become increasingly difficult for content producers to aggregate a sufficiently large audience to support the creation of high-quality content. Syndication enables content creators to amortize the cost of producing content by licensing it across multiple publishers or by maximizing the distribution of advertising-supported content. A potential drawback for content creators, however, is that they can lose control over the presentation of their content when they syndicate it to other

parties.

Distribution partners benefit by receiving content either at a discounted price, or free. One potential drawback for publishers, however, is that because the content is duplicated at other publisher sites, they cannot have an "exclusive" on the content.

For users, the fact that syndication enables the production and maintenance of content allows them to find and consume content on the Internet. One potential drawback for them is that they may run into duplicate content, which could be an annoyance.

E-commerce

[edit]
See also: E-commerce

Web syndication has been used to distribute product content such as feature descriptions, images, and specifications. As manufacturers are regarded as authorities and most sales are not achieved on manufacturer websites, manufacturers allow retailers or dealers to publish the information on their sites. Through syndication, manufacturers may pass relevant information to channel partners.[8] Such web syndication has been shown to increase sales.[9]

Web syndication has also been found effective as a search engine optimization technique.[10]

See also

[edit]

- RSS
- Atom (web standard)
- Broadcast syndication
- Content delivery platform
- Feed icon
- hAtom
- List of comic strip syndicates
- List of streaming media systems
- Print syndication
- Protection of Broadcasts and Broadcasting Organizations Treaty
- Push technology
- Software as a service
- Usenet

References

[edit]

1. ^ *Hammersley, Ben (2005). Developing Feeds with RSS and Atom. Sebastopol: O'Reilly. ISBN 0-596-00881-3.*
2. ^ "Offers Plan to Syndicate Programs." The New York Times. 12 Oct 1924: Special Features Radio Automobiles Page 14
3. ^ Broadcast syndication
4. ^ Museum of Broadcast Communications Syndication Archived 9 October 2009 at the Wayback Machine
5. ^ *Lash, Alex (3 October 1997). "W3C takes first step toward RDF spec". Archived from the original on 13 July 2012. Retrieved 16 February 2007.*
6. ^ *"Internet Content Syndication: Content Creation and Distribution in an Expanding Internet Universe" (PDF). Internet Content Syndication Council. May 2008.*
7. ^ Netcraft.com "Web Server Survey."
8. ^ Forrester Research "Must Haves for Manufacturer Web Sites"
9. ^ Internet Retailer More product content equals more sales at eCost.com
10. ^ How to Increase Your Search Ranking Fresh Business Thinking

External links

[edit]

- Image not found or type unknown Media related to Web syndication at Wikimedia Commons

- **v**
- **t**
- **e**

Web syndication

History

Blogging
Podcasting
Vlogging
Web syndication technology

Types

- Art
- Bloggernacle
- Classical music
- Corporate
- Dream diary
- Edublog
- Electronic journal
- Fake
- Family
- Fashion
- Food
- Health
- Law
- Lifelog
- MP3
- News
- Photoblog
- Police
- Political
- Project
- Reverse
- Travel
- Warblog

Technology	**General**	○ BitTorrent ○ Feed URI scheme
	Features	○ Linkback ○ Permalink ○ Ping ○ Pingback ○ Reblogging ○ Refback ○ Rollback ○ Trackback
	Mechanism	○ Thread ○ Geotagging ○ RSS enclosure ○ Synchronization
	Memetics	○ Atom feed ○ Data feed ○ Photofeed ○ Product feed ○ RDF feed ○ Web feed
	RSS	○ GeoRSS ○ MRSS ○ RSS TV
	Social	○ Inter-process communication ○ Mashup ○ Referencing ○ RSS editor ○ RSS tracking ○ Streaming media
	Standard	○ OPML ○ RSS Advisory Board ○ Usenet ○ World Wide Web ○ XBEL ○ XOXO

- Audio podcast
- Enhanced podcast
- Mobilecast
- Narrowcasting
- Peercasting
- Screencast
- Slidecasting
- Videocast
- Webcomic
- Webtoon
- Web series

Form

- Anonymous blogging
- Collaborative blog
- Columnist
- Instant messaging
- Liveblogging
- Microblog
- Mobile blogging
- Spam blog
- Video blogging
- Motovlogging

Media

Alternative media

- Carnivals
- Fiction
- Journalism
  - Citizen
  - Database
- Online diary
- Search engines
- Sideblog
- Software
- Web directory

Micromedia

- Aggregation
  - News
  - Poll
  - Review
  - Search
  - Video
- Atom
- AtomPub
- Broadcatching
- Hashtag
- NewsML
  - 1
  - G2
- Social communication
- Social software
- Web Slice

Related

- Blogosphere
- Escribitionist
- Glossary of blogging
- Pay per click
- Posting style
- Slashdot effect
- Spam in blogs
- Uses of podcasting

Check our other pages :

- Web Design Parramatta
- SEO Parramatta
- Website designers Parramatta
- Affordable SEO Parramatta
- SEO consultant Parramatta
- Custom web design Parramatta
- Parramatta digital marketing

Custom web design Parramatta

SEO Parramatta

Phone : 1300 684 339

City : Sydney

State : NSW

Zip : 2000

Google Business Profile

Google Business Website

Company Website : https://sydney.website/seo-sydney/local-seo/seo-parramatta/

USEFUL LINKS

SEO Website

SEO Services Sydney

Local SEO Sydney

SEO Ranking

SEO optimisation

LATEST BLOGPOSTS

SEO community

SEO Buzz

WordPress SEO

SEO Audit

Sitemap

Privacy Policy

About Us

SEO Castle Hill | SEO Fairfield | SEO Hornsby | SEO Liverpool | SEO North Sydney | SEO Norwest | SEO Parramatta | SEO Penrith | SEO Strathfield | SEO Wetherill Park

Follow us