On the Feasibility of Geographically Distributed Web Crawling

(Invited Paper)

B. Barla Cambazoglu
Yahoo! Research
Barcelona, Spain
barla@yahoo-inc.com

Flavio Junqueira
Yahoo! Research
Barcelona, Spain
fpj@yahoo-inc.com

Vassilis Plachouras
Yahoo! Research
Barcelona, Spain
vassilis@yahoo-inc.com

Luca Telloli
Yahoo! Research
Barcelona, Spain
telloli@yahoo-inc.com

ABSTRACT
We identify the issues that are important in design of a geographically distributed Web crawler. The identified issues are discussed from a "benefit" and "challenge" point of view. More specifically, we focus on the effect of geographical locality of Web sites on crawling performance, and, as a practical study, investigate the feasibility of a distributed crawler in terms of network costs. For this purpose, we conduct various experiments to collect network access statistics about the servers in the educational domains of eight different countries (USA, Canada, Chile, Brazil, Spain, Portugal, Turkey, and Greece). We gather the statistics from four different sites located in USA, Brazil, Spain, and Turkey using echoping. The results favor geographically distributed Web crawling in terms of crawling throughput.

Keywords
Distributed Web crawling, spatial locality, throughput.

1. BACKGROUND
The enormous size of the content on the Web, its very fast expansion rate, and the high frequency of page modifications render the Web crawling problem [15, 29] as one of the biggest challenges in search engine design. The importance of the problem increases given the fact that the user-perceived quality of a search engine heavily depends on the content obtained by its crawling component. A promising solution to this problem is geographically distributed Web crawling, which has not attracted much research attention, perhaps, due to the practical challenges in development. In this paper, our aim is to illustrate the benefits of geographically distributed Web crawling with some supportive performance results and to present the challenges involved.

INFOSCALE 2008, June 4-6, Vico Equense, Italy
Copyright © 2008 978-963-9799-28-8
DOI 10.4108/ICST.INFOSCALE2008.3550

In Web crawling, the main purpose is to traverse the Web pages by using the hyperlink structure and download as many pages as possible under certain performance and quality objectives. In a typical crawling system, one or more crawlers start from a fixed set of seed pages and, by parsing those pages' content, discover URLs to other pages in the Web space. Discovered URLs are added to a queue, from which they are extracted in a certain order and corresponding pages are downloaded from the Web. URL extraction and enqueueing steps are repeated on the newly found content in an iterative manner. The entire process is performed in sessions, in which case a new crawl is started from scratch after a while, or the process is repeated infinitely, which requires periodically re-downloading previously crawled pages.

Web crawling is a computationally expensive process that demands large amount of resources, including CPU, disk storage, memory, and network. Consequently, if the target is to crawl a significant portion of the Web, the single-processor systems will fail to achieve this. One immediate architectural solution is to employ parallelization and perform crawling on multiple processors. This way, the performance bottlenecks on CPU and memory can be relaxed and fairly good speedups can be achieved. However, it is not possible to obtain the maximum benefit and achieve scalability in terms of network if all processors are located at a single data center. On the other hand, geographically distributed Web crawling (from now on, distributed crawling for short), where multiple parallel crawlers are located at geographically distant data centers, has the potential benefit that information about the spatial locality of the Web servers can be used to improve the download times, achieving scalability on the network resource as well.

The organization of the paper is as follows. In Section 2, we summarize some important concepts in crawling, briefly explaining their roles in the context of distributed crawling. This section also provides pointers to previous works on crawling in the literature. We illustrate some benefits that can be achieved by distributed crawling in Section 3. Section 4 discusses the challenges involved. In Section 5, the results of various network experiments are presented. The paper is concluded in Section 6 with some open issues.

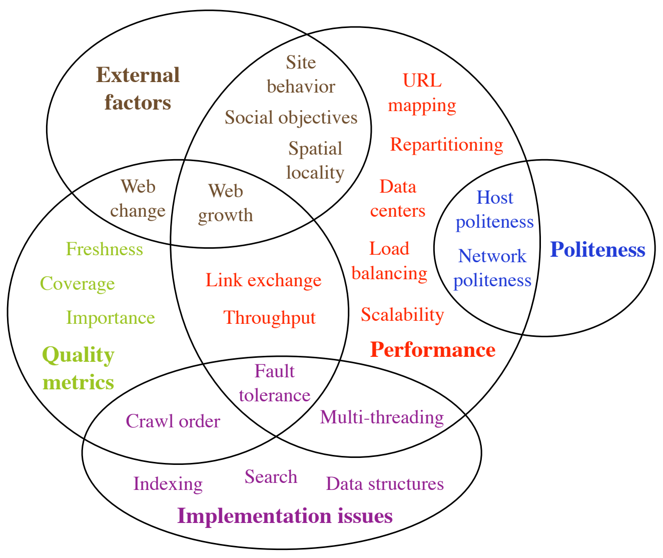


Figure 1: Various concepts in Web crawling and their classification under different headings.

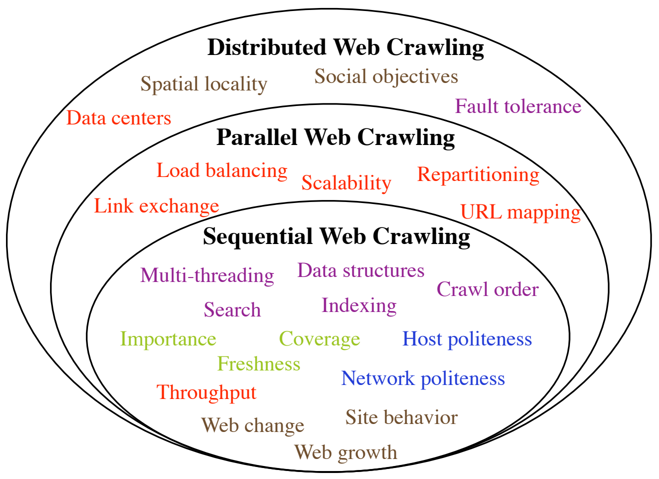


Figure 2: The architectural classification of the Web crawling concepts in Fig. 1.

2. CONCEPTS

The purpose of this section is to go over the most important concepts in Web crawling research and practice, identifying the ones relevant to distributed crawling. For this purpose, we base the discussion in this section on Figs. 1 and 2. Fig. 1 classifies the most important concepts in Web crawling under five main headings: quality metrics, external factors, performance, implementation issues, and politeness. The color coding is used to indicate the primary class for concepts, which may appear under multiple headings. Fig. 2 classifies the same set of concepts in Fig. 1 according to their importance to a particular type of architecture, namely, sequential, parallel, and distributed crawling. Note that all concepts classified under sequential crawling are also related to parallel and distributed crawling, and the same relation holds between parallel crawling and distributed crawling.

2.1 Quality Metrics

In evaluating the quality of a crawler, in general, there are three important metrics: the crawler's coverage of the Web [24], the freshness of the pages in its repository [8, 9, 11, 27], and the importance of the downloaded pages [1] from the user or query processing perspective. The coverage refers to the percent of the Web discovered or downloaded by the crawler. This is an important metric in that it directly affects the recall of the search engine. Page freshness is usually a measure of how outdated the local copy of a page is relative to the page's original copy on the Web. Freshness of a repository is said to increase as it contains fewer outdated pages or pages are less outdated. Increasing page freshness is important to be able to provide the most up-to-date information on the Web to the users. Finally, the importance metric relates to increasing the percent of important or popular pages in the repository. This may help precision of a search engine.

In assessing the quality of a distributed crawling system, the above-mentioned set of quality metrics can be used without any modification. However, depending on the techniques employed, the quality of a distributed crawler may degrade compared to that of a sequential crawler. In this respect, the problems that stand for parallel crawlers [10] also apply to distributed crawling [4, 30, 33].

2.2 External Factors

The two major external factors that affect the quality metrics are Web growth [5] and Web change [5, 8], which have inverse effects respectively on coverage and freshness. Increasing the hardware resources is the main method used in practice to achieve sustainable consistency in coverage and freshness of a crawling system. Another important external factor is the behavior of Web sites. Behaviors of certain types of sites may make crawling process more difficult. This includes hostile sites (e.g., spider traps, infinite domain name generators), spam sites (e.g., link farms [18, 19, 20]), sites with restricted content (e.g., robot exclusion [23]), and unstable sites (e.g., variable host performance, unreliable networks). These four types of sites affect distributed crawling in the same way they affect sequential or parallel crawling, perhaps, with the exception of sites with unreliable networks. This is because, in distributed crawling, having crawlers closer to Web sites may increase the chance of avoiding intermediate network failures between the crawlers and sites. We will elaborate more on this in Section 3.

The remaining two external factors, i.e., social objectives and spatial locality [7] may have an impact on the performance of a distributed crawler. In this context, the social objective refers to a specific target content type in crawling [13]. For example, the target content may be the Web sites that contain pages written in a specific language (e.g., Spanish, Russian), or are hosted in a specific set of countries (Spain, Russia) or a region (e.g., Middle East). If there is a social objective, the placement of crawlers becomes important and should conform as much as possible with the placement of the target content. The spatial locality refers to the geographical placement and closeness of Web sites to crawlers [16, 17]. Note that this is different from the country-specific content objective as belonging to a country does not always guarantee low spatial proximity (e.g., very large or disconnected countries) and spatial proximity does not always guarantee belonging to the same country (e.g., sites near the country borders). Spatial locality can be very important for a distributed crawler and may provide advantages over parallel crawlers. We will discuss this in more detail in Section 3 and will provide some practical performance results in Section 5, illustrating importance of spatial locality in crawling.

2.3 Performance

Among the performance concepts given in Fig. 2, throughput is the only one applicable for all types of crawling. Throughput is a measure of the amount of Web content downloaded by the crawler per unit of time. High throughput is desirable as it affects all quality metrics, e.g., a high-throughput crawler may have a better coverage or may keep its repository fresher. Among the performance concepts, load balancing and scalability are very well-known performance metrics in the parallel computing domain. In the crawling context, they have similar meanings. Load balancing refers to assigning equal amounts of workload to crawlers [6]. Here, workload can be defined by the number of pages or bytes to download, the amount of content stored, the crawling time, or a combination of these. Load balancing may be more important for distributed crawling than parallel crawling since there are social objectives that imply variation on the target content sizes of crawlers. Moreover, it is more likely to have heterogeneity in hardware/network resources of different data centers. Scalability refers to the performance change in the system when a new crawler is added. In parallel crawling, the bandwidth is most likely to be the first resource to limit scalability. Due to this constraint, after a certain number, adding new crawlers do not improve overall performance; in contrast, bottlenecks may be observed at other parts of the system, leading to poor performance.

The remaining performance concepts in Fig. 2 are more specific to the Web crawling problem. In parallel crawling, preventing the crawlers from downloading the same content and maximizing the Web coverage requires exchange of links between the crawlers. The volume of communication during the link exchange may become a performance issue if the ratio of inter-crawler communication times to page download times is high (e.g., distributed crawling, intra-site crawling). The link exchange scheme requires mapping of URLs to crawlers [6, 31]. In parallel crawlers, the mapping is typically performed in a randomized manner using MD5 hashes of URLs. In distributed crawling, on the other hand, more intelligent mapping is necessary as this mapping has a direct effect on the overall throughput of the system. Note that external factors like spatial locality and social objectives make it harder to find a simple solution to the problem. Two other concepts closely related to URL mapping are repartitioning and data centers. In distributed crawling, repartitioning is required due to the Web growth/change, the changes in geographical placement of Web sites, and variations in social objectives. Periodical repartitioning provides a better URL mapping, but it brings an overhead as well since this may necessitate moving certain portions of the crawled data among the crawlers. We will go into more depth on URL mapping and repartitioning in Section 4, where we discuss the challenges. Finally, the data center concept refers to the number and placement of data centers. As we will discuss in Section 3, it may be possible to increase the accessibility of Web sites by means of distributed crawling.

2.4 Implementation Issues

A wide variety of implementation issues in practice may affect both the quality and performance of a crawler. For example, multi-threading is employed for better utilization of the network to achieve higher throughput values. However, as in scalability, the network is the bottleneck before multi-threading, i.e., there is an optimum number of threads that provides the best performance gain, depending on various parameters. Crawling order is the decision about how the download of pages should be prioritized [3, 12, 26]. This decision has consequences on the page importance metric. The vast nature of the crawling problem requires very careful selection of internal data structures [21, 25]. Among the most important data structures are the queue of the URLs to be downloaded (i.e., the frontier), the list of the URLs seen, the local DNS cache, and the cache of robots.txt files. Out-of-core data processing strategies must be employed to manage these structures as they grow very quickly and do not fit into the memory.

Availability of a crawler is important for maintaining a steady performance. Availability requires fault tolerance, i.e., continuous execution against external or internal, software or hardware problems. Fault tolerance is relatively easy to achieve in parallel crawling, e.g., by assigning URLs of a failed processor to a backup processor. However, in distributed crawling, this may become an issue as the workload of a problematic/inaccessible data center must be shared by the others [4]. This requires repartitioning of the URL space among crawlers which enforces remapping of some URLs. Note that, despite its overhead, the availability on the larger scale will be higher in distributed crawling than parallel crawling.

There are also design and implementation issues concerning indexing and search [28], which are two components that are natural successors to the crawling process. In a fully or partially distributed search engine, there are design issues in sharing the data collected by the crawlers with the rest of the system. These involve issues about the connectivity and interaction of the crawling, indexing, and search components.

2.5 Politeness

If it is not reined in, the execution of a crawler may incur a burden on Web servers, eventually causing them to consider the crawler as a hostile program. Hence, crawlers should be as polite as possible to Web servers [14], trying not to overload them, without sacrificing throughput. Typically, they do this by limiting the number and duration of open network connections to servers and also obeying the robot exclusion protocol. Network politeness is an extension of the host politeness concept to the network. Crawlers should disperse the load across the network as much as possible, avoiding to create hot spots within local networks or sub-networks.

3. BENEFITS

We identify five distinct benefits for distributed crawling in the following areas: crawling throughput, network politeness, Web coverage, availability, and coupling of distributed crawling with distributed indexing and search. In what follows, we elaborate more on these benefits.

A distributed crawler has the potential to use network resources more efficiently than a centralized crawler. Consider a distributed crawling system, where each crawler is responsible for downloading Web pages that are stored on servers geographically close to itself [16, 17]. In such a scenario, the average network latency for downloading Web pages is expected to be smaller if compared to the average latency of a scenario where Web pages are downloaded by a central crawler. A further benefit of spatial locality may be on the connection timeout values that crawlers set against non-responsive servers. These values now may be set to lower values, reducing the time spent on servers with connection problems. The choice of geographical distance rather than logical network distance, such as IP path length or autonomous system path length, relates to findings in [22], suggesting that round-trip time correlates better with geographical distance. In general, spatial locality implies higher throughput for the overall crawling process. In Section 5, we present experimental results that support this claim.

A conceptually different benefit that is again related to the network usage is the improved politeness to the network. This benefit stems from the fact that, in distributed crawling, the network traffic generated by a crawler is expected to follow shorter routes. Hence, the overhead on routers and switches is likely to be less and affect smaller parts of the network. For example, if a crawler in North America only crawls Web pages hosted on servers in the same geographic region, then the generated traffic is less likely to be routed through Europe or other geographic regions. There will be cases, however, where the traffic would be routed through a geographically remote network. In such cases, it may be more appropriate to use logical network distance metrics rather than the geographic distance to predict the latency [22].

Distributed crawling is potentially more resilient to network partitions, which cannot be ignored at the scale of the Internet [32]. If crawling is performed in a centralized manner, then a network partition may render a significant part of the Web inaccessible and hence reduce the coverage or freshness of the crawler. On the other hand, depending on where the crawlers are running, a network partition may affect fewer or no crawlers. For example, in the case of a network partition, a distributed crawler may still be able to download Web pages, but only the communication among the crawlers to exchange discovered links may be disrupted. Moreover, being closer to Web servers may increase to probability of avoiding the failures in sub-networks.

Another benefit of a distributed crawler is increased availability. A catastrophic event, a natural disaster, or a network problem may render a centralized crawler inoperative, which, in turn, severely affects the content quality of the search engine. From a business continuity point of view, distributing crawling over multiple data centers may provide some form of fault tolerance as the workload of an inoperative center may be shared by the others. Even if the workload is not shared, the remaining crawlers can continue to crawl, at least, their portions of the Web.

In the context of a distributed Web search engine [2], distributed crawling could be a natural design choice. Suppose that a distributed search engine consists of a set of geographically distributed search centers, where each center has a local index of a subset of Web pages and processes user queries from the same geographic region. If the search engine uses a centralized crawler, then it is necessary to index the pages and partition the resulting data structures where the crawler stores the downloaded pages. Alternatively, each search center can build a local index by fetching its subset of the crawled pages in a compressed form. In both cases, a substantial amount of data needs to be transferred between the crawling center and search centers. With distributed crawling, however, it is possible to minimize or completely eliminate this overhead if the crawlers are placed closer or inside the search centers.

4. CHALLENGES

As current challenges for distributed crawling, we identify four research problems in the following topics: Web partitioning/repartitioning, data center placement, link classification, and coupling of crawling with indexing and search. In what follows, we discuss these research problems.

In general, the overhead of a crawler is dominated by the network costs. An important challenge in distributed crawler design is the problem of finding the optimum mapping of Web servers to crawlers, i.e., a Web partition that minimizes the overheads of the distributed crawler. There are mainly three types of overheads: page download times, the overhead of link exchange between the crawlers, and the overhead due to the partitioning process itself. The solution to this partitioning problem should consider all of these overheads. The first objective in such a solution should be to partition the Web in such a way that the page download times of the crawlers are as low as possible. Therefore, the solution should take into account the geographical proximity of servers to crawlers. Additional techniques may include crawling distant but lightly loaded servers/networks, taking time zones of servers into consideration, and utilizing previously collected network statistics. The second objective should be to reduce the communication overhead during the exchange of inter-crawler links as, depending on the number and connectivity of the crawlers, this communication may incur a significant overhead. Finally, the third objective should be to keep the difference between the old and new partitions as low as possible since remapping may require relocating certain portions of the data among different centers.

Due to fault tolerance requirements or changes in various constraints, the above-mentioned partitioning process may need to be repeated. For example, in a distributed search engine, crawlers may be matched to local search centers and hence be constrained to download the pages that will be served by the local search engine they match. Load balancing may be another constraint as the size of the target contents as well as the resources in different data centers may vary. The ultimate solution to this partitioning problem would combine all these constraints and objectives, requiring complex repartitioning algorithms that are NP-hard [6].

A related, but completely reverse problem is, given a set of Web servers and a set of constraints and objectives, to find the optimum geographical placement for a fixed number of data centers. In this problem, most constraints and objectives are similar to the ones in the above-mentioned Web partitioning problem. However, the problem is different than the Web partitioning problem since the geographical coordinates of the data centers is no longer a constraint, but a variable that we try to optimize. In the case of distributed search, placement of search centers may be another constraint for this problem. Also, the site accessibility and network politeness benefits mentioned in Section 3 may be additional objectives. A variation of this problem is the case where the number of data centers is also not known. Note that there must be a bound on the optimum number of centers as link exchange and repartitioning overheads would prevent scalability.

A problem related to exchange of links among the crawlers is classification of the sites pointed by the links. If partitioning is performed under the constraint of certain social objectives, mechanisms must be developed to identify, for example, geographical regions or languages of sites. This can be done by utilizing the information available in the site content, connectivity of the site, or IP databases. Sites with multiple languages form another issue as it is not clear how these sites should be assigned to the crawlers. Furthermore, strategies are required to handle newly discovered sites, which are more difficult to associate with a crawler as not enough information is available about the site.

Finally, a design challenge exists in coupling of distributed crawling with distributed indexing and distributed search. For example, the data obtained by a distributed crawler may need to be moved to a single data center, replicated on different data centers, or partitioned among a number of data centers. Decisions must be given on what data to move (e.g., pages or index), how to move (i.e., compression), and how often to move (i.e., synchronization). Based on the way the index is distributed, search could be centralized or distributed. Distributed crawling may couple well with a partitioned index and hence distributed search. Cost models should be developed and feasibility should be investigated for different coupling scenarios.

5. EXPERIMENTS

We conducted various experiments to illustrate the performance gains achievable by distributed crawling in practice. For this purpose, we continuously downloaded the root pages of a fixed set of Web sites from a fixed set of crawling nodes over a certain period of time and collected network access statistics. These statistics are used to estimate and compare simulated throughputs of different crawling configurations.

5.1 Setup

In the experiments, we use four separate crawlers located in different countries (Brazil, USA, Spain, and Turkey), belonging to different continents (South America, North America, Europe, and Asia, respectively). The crawlers download pages from eight target countries (Brazil (BR), Chile (CL), USA (US), Canada (CA), Spain (SP), Portugal (PT), Turkey (TR), and Greece (GR)) that host the Web sites. The countries in this set are selected such that they are geographically neighbor to at least one country that runs a crawler. The Web sites in the target countries are selected from the educational domain of each country. The rationale behind the selection of educational sites is as follows:

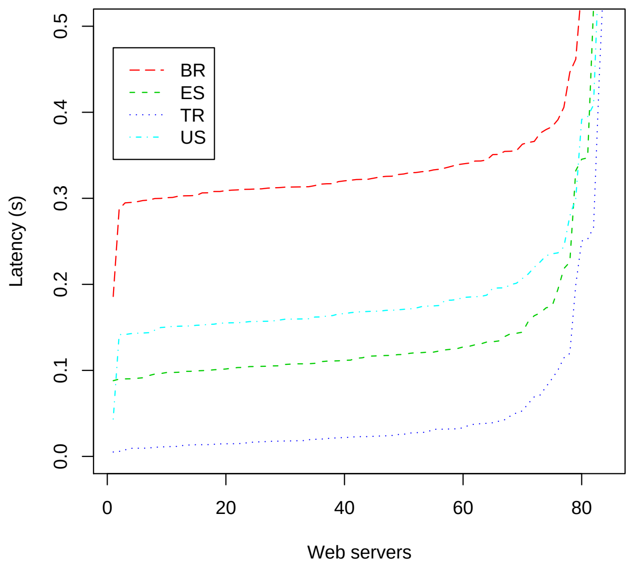
1. These sites are easier to identify by their top level domain names,
2. It is highly likely that they are physically located in the country that their top level domain name indicates,
3. There is less variation in server loads compared to the sites in other domains.

Table 1: Information about the number of sites and the average root page sizes of the servers in the `.edu` domains of the target countries

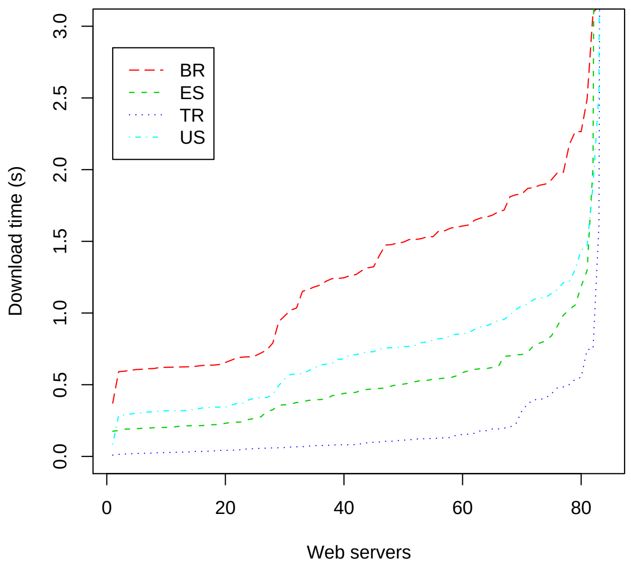
Crawled country	Avg. page size (byte)	# of crawled sites	Total # of sites
BR	16,825	98	1582
CA	15,012	99	198
CL	15,617	76	78
ES	16,298	96	174
GR	13,369	60	64
PT	15,095	99	166
TR	20,726	85	87
US	20,177	100	3352

Table 1 displays, in the two rightmost columns, our sample size of the `.edu` sites and the total number of `.edu` sites for the selected countries, as reported by `http://www.webometrics.info`. For the countries with more than 100 sites, we used a randomly selected subset of 100 sites to obtain some balance on sampling spaces. The sites that are not accessible by at least one crawler at any time during the lifetime of the experiments are further removed. In the table, the second column reports the average root page size of the servers for each target country. These sizes are obtained by averaging the sizes of the collected root page samples.

To obtain the network statistics, we use our customized version of the echoping software (the original is available at `http://echoping.sourceforge.net`). Although the primary purpose of this software is to provide a ping facility for different network protocols, it also facilitates the download of HTML pages. In order to minimize the temporal variations in host and network loads, the experiments were conducted over a week, downloading pages from the target sites repetitively in a randomized order. In order to enforce some politeness, a delay of one second is placed between any two consecutive page requests. Echoping is run with the -A option, allowing generated HTTP requests to skip intermediate caches on proxies and gateways. Unlike typical crawlers, echoping is a single-threaded process, manually



(a) HTTP latency



(b) Download time

Figure 3: HTTP latencies and download times observed from four different countries in crawling Turkish Web sites.

programmed to crawl a fixed set of URLs; therefore, the link exchange is not considered in our experiments.

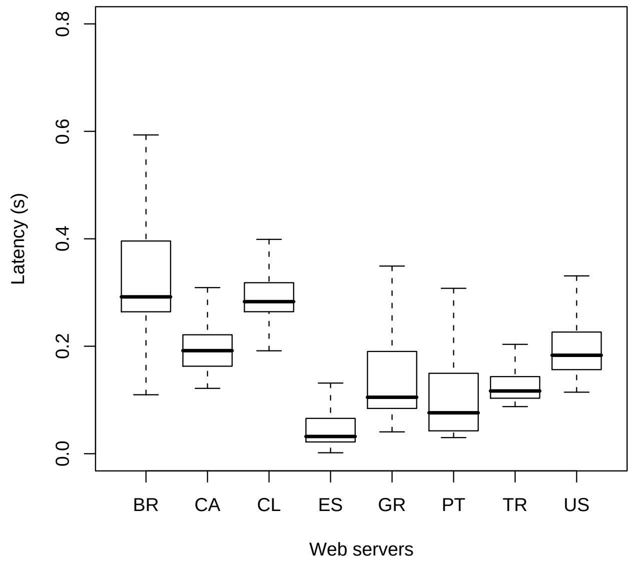
The samples collected by echoping include the total time spent by a crawler in downloading the root page of a Web site. The download time can be subdivided into four subcomponents: DNS resolution time, TCP latency, HTTP latency, and page transfer time. In our experiments, the DNS resolution time is negligible since all host names are translated to their corresponding IPs beforehand. The TCP latency is the time spent for the three-way handshake between a crawler and a server to open a connection. The HTTP latency is the time between a `GET` request is issued by the crawler and the first byte of the server response is received. The transfer time is the time to transfer the document from the remote Web server to the crawling process. The total download time we measure includes the TCP and HTTP latencies as well as the transfer time of the pages. In our experiments, in general, the TCP and HTTP latencies show a correlation and are very close.

5.2 Network Performance

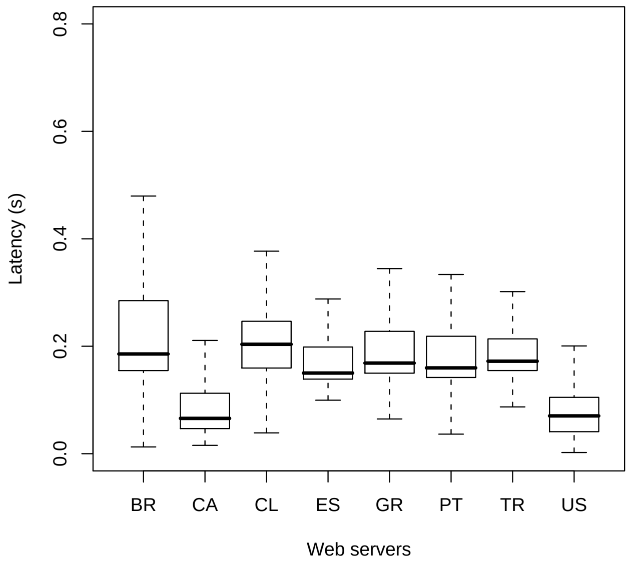
Fig. 3(a) displays the medians of HTTP latencies over a number of trials in crawling Turkish Web sites from different geographical locations. All latency values are sorted for each crawler in increasing order. As expected, the latencies are much lower for the local Turkish crawler, which is followed by the Spanish crawler. The highest latencies are observed when crawling from Brazil. A sharp increase is observed in latencies for the servers after around the 75th server. This could be due to highly-loaded or slow servers or misconfigured network equipment on those servers.

In terms of overall crawling performance a similar behavior is observed in Fig. 3(b), where the medians of download times are given. This figure can be investigated in three parts. For about the first 20 servers, the download time is almost constant (in fact, almost equal to the sum of TCP and HTTP latencies). This is due to the Web pages that are simple redirects to other pages, pages that return various error messages (e.g., 404 errors), or pages with very little content in their body. For such pages, the HTTP response contains just a header; therefore, the transfer time is almost negligible. The download time for servers between 20 and 80 show a linear increase with increasing page size. The download times for a few servers at the end of the axis show a more variable behavior. The reason for this could be overloaded sub-networks, misconfigured networks, or overloaded servers, which is hard to identify.

Fig. 4(a) and Fig. 4(b) respectively display the HTTP latencies for the two crawlers located in Spain and the United States. The results indicate that there is a significant latency gap between the crawl of geographically close versus geographically distant sites. The latency values obtained for the Brazilian and Turkish crawlers (not given in the paper) correlate more with those of the Spanish crawler in that the variation is higher. The United States forms an exception since the median values for different target countries tend to be much closer to each other compared to those of other crawling countries. In particular, medians for all target countries, except for the United States and Canada, range around a latency of 0.2 second. This shows the technological advantage of network resources in the United States and partly justifies the choice of the United States for a centralized crawling architecture. Fig. 5(a) and Fig. 5(b) pro-

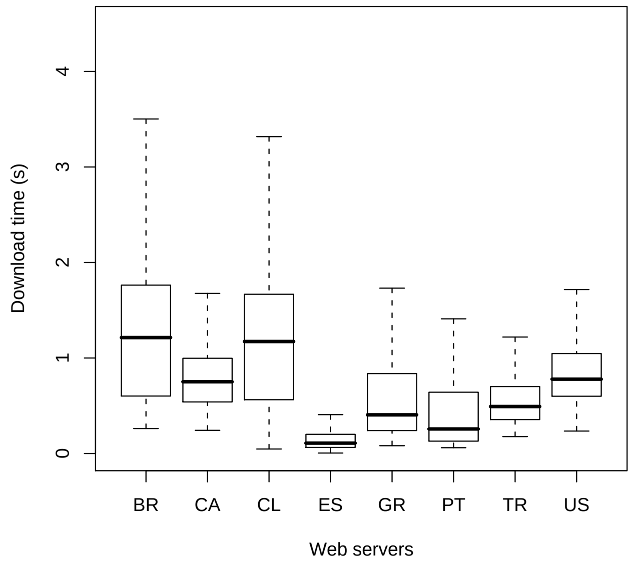


(a) ES crawler

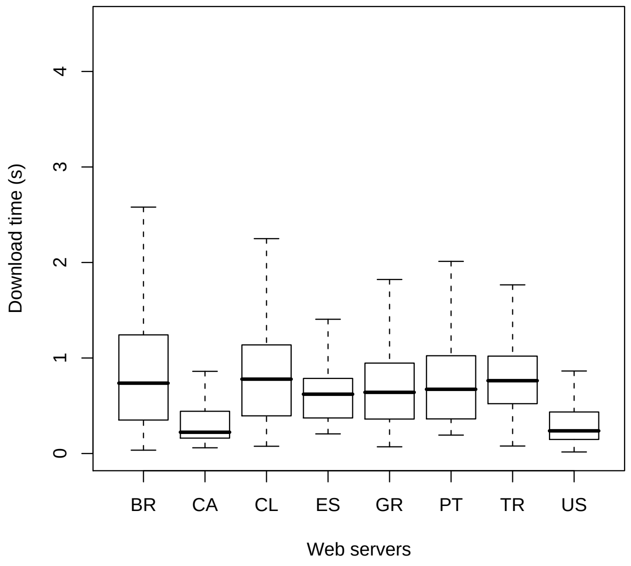


(b) US crawler

Figure 4: HTTP latencies for two different crawlers.



(a) ES crawler



(b) US crawler

Figure 5: Download times for two different crawlers.

Table 2: Medians of download times (in seconds) for different pairs of crawling countries (in rows) and target countries (in columns)

	BR	CA	CL	ES	GR	PT	TR	US
BR	**0.3324**	1.1425	**0.5210**	0.9556	1.0218	0.9701	1.3490	0.9125
ES	1.1033	*0.7017*	1.0907	**0.1021**	**0.3858**	**0.2152**	*0.4678*	*0.7597*
TR	1.3447	0.7638	1.3359	*0.4069*	*0.3896*	*0.4484*	**0.1043**	0.8764
US	*0.6842*	**0.2119**	*0.7412*	0.5657	0.5959	0.5592	0.7208	**0.2318**

vide similar information for the total page download times. These results, in general, correlate well with those in Fig. 4.

Table 2 gives a broader view of the results from our experiments, showing the median download times for all pairs of crawlers (in rows) and target countries (in columns). We show the lowest medians achieved for a target country in bold and the second lowest medians in italics.

5.3 Crawling Throughput

In this section, we estimate the throughput of various crawling scenarios using the network statistics provided in the previous section. We assume that we have a set $\mathcal{C}$ of geographically distributed crawlers and a set $\mathcal{U}$ of URLs to be crawled. Given a subset $\mathcal{U}(C) \subset \mathcal{U}$ of URLs that crawler $C \in \mathcal{C}$ is responsible for crawling, the throughput $T(C)$ of crawler C can be calculated as

$$T(C) = \frac{\sum_{U \in \mathcal{U}(C)} P(U)}{\sum_{U \in \mathcal{U}(C)} D(C,U)}, \qquad (1)$$

where $U \in \mathcal{U}$ is a URL, $P(U)$ is the size of the page pointed by U, and $D(C,U)$ is the time spent by C for downloading U. In this equation, it is possible to estimate $D(C,U)$ as

$$D(C,U) = L(C,U) + H(C,U) + P(U)/B(C,U), \qquad (2)$$

where $L(C,U)$, $H(C,U)$, and $B(C,U)$ respectively represent the TCP latency, HTTP latency, and available bandwidth between crawler C and the server hosting U. Here, we assume that a new TCP connection is established for each page request and only one page is requested from the server per request. The aggregate throughput of crawlers in set $\mathcal{C}$ can be calculated as

$$T(\mathcal{C}) = \sum_{C \in \mathcal{C}} T(C), \qquad (3)$$

assuming crawlers operate continuously.

To make a comparison between different scenarios, we report results for seven crawling configurations using subsets of the crawler set $\mathcal{C} = \{\mathrm{BR, ES, TR, US}\}$: $\mathcal{C}_{\mathrm{BR}} = \{\mathrm{BR}\}$, $\mathcal{C}_{\mathrm{US}} = \{\mathrm{US}\}$, $\mathcal{C}_{\mathrm{ES}} = \{\mathrm{ES}\}$, $\mathcal{C}_{\mathrm{TR}} = \{\mathrm{TR}\}$, $\mathcal{C}_{\mathrm{BR,TR}} = \{\mathrm{BR, TR}\}$, $\mathcal{C}_{\mathrm{ES,US}} = \{\mathrm{ES, US}\}$, and $\mathcal{C}_{\mathrm{ALL}} = \{\mathrm{BR, ES, TR, US}\}$. The complete URL space for the countries we used in the experiments is $\mathcal{U} = \{\mathrm{BR, CL, US, CA, PT, ES, TR, GR}\}$, where each set element denotes the set of URLs in the corresponding country. For the first four crawler configurations, we use $\mathcal{U}(C_{\mathrm{BR}}) = \mathcal{U}(C_{\mathrm{US}}) = \mathcal{U}(C_{\mathrm{ES}}) = \mathcal{U}(C_{\mathrm{TR}}) = \mathcal{U}$, that is, a single centralized crawler covers the whole URL space. For the $\mathcal{C}_{\mathrm{BR,TR}}$ and $\mathcal{C}_{\mathrm{ES,US}}$ configurations, we use $C_{\mathrm{BR}} = \{\mathrm{BR, CL, US, CA}\}$, $C_{\mathrm{TR}} = \{\mathrm{GR, TR, ES, PT}\}$ and $C_{\mathrm{ES}} = \{\mathrm{ES, PT, GR, TR}\}$, $C_{\mathrm{US}} = \{\mathrm{US, CA, BR, CL}\}$. Basically, we have two distributed crawling configurations consisting of two crawlers, where each crawler is responsible for crawling one side of the Atlantic. For the last configuration, we use $\mathcal{U}(C_{\mathrm{BR}}) = \{\mathrm{BR, CL}\}$, $\mathcal{U}(C_{\mathrm{US}}) = \{\mathrm{US, CA}\}$, $\mathcal{U}(C_{\mathrm{ES}}) = \{\mathrm{ES, PT}\}$, and $\mathcal{U}(C_{\mathrm{TR}}) = \{\mathrm{TR, GR}\}$. This configuration represents a distributed crawling system with four crawlers, each responsible for the URLs of the local and a neighbor country.

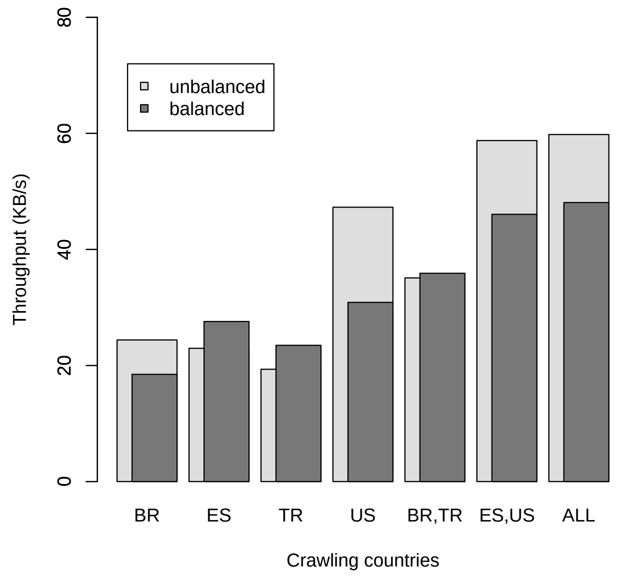


Figure 6: Throughput comparison between various crawling configurations.

Fig. 6 compares the throughput for the crawling configurations given above. In computing the values in this figure, we assume that the total amount of crawling resources available in each configuration is the same. Therefore, we divide the throughput values computed using (3) by the number of crawling countries in the configuration. In this experiment, we consider two cases for page distribution in target countries: each target country having an equal amount of Web content versus and unbalanced content distribution. In Fig. 6, the light gray and dark gray bars represent the throughputs achieved for the unbalanced and balanced cases, respectively. For the unbalanced distribution, we use the real-life distribution of the `.edu` sites in countries, shown in Table 1.

According to Fig. 6, among the centralized crawling configurations, the US crawler is the best performer for both balanced and unbalanced cases. For the balanced case, the Spanish crawler is the second best due to its advantage of being geographically close to other countries. For the unbalanced case, the second best is the Brazilian crawler due to the large amount of Web content available in Brazil. Among the distributed crawling configurations, the $\mathcal{C}_{\mathrm{ES,US}}$ configuration performs considerably better than the $\mathcal{C}_{\mathrm{BR,TR}}$ configuration. The $\mathcal{C}_{\mathrm{BR,TR}}$ configuration performs even worse than the $\mathcal{C}_{\mathrm{US}}$ configuration for the unbalanced case. It is interesting to note that the throughput gap between the $\mathcal{C}_{\mathrm{ES,US}}$ and the $\mathcal{C}_{\mathrm{ALL}}$ configurations is marginal.

6. CONCLUSION

A successful feasibility analysis on distributed crawling requires further work in both theory and practice. On the theoretical side, appropriate cost models should be developed. These cost models should include financial costs (operational, maintenance, revenue), scalability (number of data centers, number of crawlers per center, the network bandwidths), and performance (download, link exchange, repartitioning times). On the practical side, the trends in the Web should be followed. As developing countries have more Web sites in the future, distributed crawling might be the road to choose. However, with the current distribution of Web pages among countries and the current network infrastructure, centralized crawling from a center located in the United States still seems to be more feasible.

7. REFERENCES

[1] S. Abiteboul, M. Preda, and G. Cobena. Adaptive on-line page importance computation. In *Proceedings of the 12th International World Wide Web Conference*, pages 280–290, 2003.

[2] R. Baeza-Yates, C. Castillo, F. Junqueira, V. Plachouras, and F. Silvestri. Challenges in distributed information retrieval. In *Proceedings of the 23rd International Conference on Data Engineering*, pages 6–20, 2007.

[3] R. Baeza-Yates, C. Castillo, M. Marin, and A. Rodriguez. Crawling a country: better strategies than breadth-first for web page ordering. In *Special Interest Tracks and Posters of the 14th International World Wide Web Conference*, 2005.

[4] P. Boldi, B. Codenotti, M. Santini, and S. Vigna. Ubicrawler: a scalable fully distributed web crawler. *Software Practice & Experience*, 34(8):711–726, 2004.

[5] B. E. Brewington and G. Cybenko. How dynamic is the web? *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 33(1-6):257–276, 2000.

[6] B. B. Cambazoglu, A. Turk, and C. Aykanat. Data-parallel web crawling models. *Lecture Notes in Computer Science*, 3280:801–809, 2004.

[7] B. B. Cambazoglu, A. Turk, E. Karaca, C. Aykanat, and T. Kucukyilmaz. Architecture of a grid-enabled search engine. *Information Processing & Management*, 43(3):609–623, 2007.

[8] J. Cho and H. Garcia-Molina. The evolution of the web and implications for an incremental crawler. In *Proceedings of the 26th International Conference on Very Large Data Bases*, pages 200–209, 2000.

[9] J. Cho and H. Garcia-Molina. Synchronizing a database to improve freshness. In *Proceedings of ACM International Conference on Management of Data*, pages 117–128, 2000.

[10] J. Cho and H. Garcia-Molina. Parallel crawlers. In *Proceedings of the 11th International World Wide Web Conference*, pages 124–135, 2002.

[11] J. Cho and H. Garcia-Molina. Estimating frequency of change. *ACM Transactions on Internet Technology*, 3(3):256–290, 2003.

[12] J. Cho, H. Garcia-Molina, and L. Page. Efficient crawling through url ordering. *Computer Networks and ISDN Systems*, 30(1-7):161–172, 1998.

[13] C. Chung and C. Clarke. Topic-oriented collaborative crawling. In *Proceedings of the 11th International Conference on Information and Knowledge Management*, pages 34–42, 2002.

[14] D. Eichmann. Ethical web agents. In *Second International Conference on World Wide Web*, pages 3–13, 1994.

[15] D. Eichmann. The rbse spider – balancing effective search against web load. In *First International Conference on World Wide Web*, pages 113–120, 1994.

[16] J. Exposto, J. Macedo, A. Pina, A. Alves, and J. Rufino. Geographical partition for distributed web crawling. In *Proceedings of the 2005 Workshop on Geographic Information Retrieval*, 2005.

[17] J. Exposto, J. Macedo, A. Pina, A. Alves, and J. Rufino. Efficient partitioning strategies for distributed web crawling. In *Proceedings of the 21st International Conference on Information Networking*, 2007.

[18] Z. Gyongyi and H. Garcia-Molina. Link spam alliances. In *Proceedings of the 31st international Conference on Very Large Data Bases*, pages 517–528, 2005.

[19] Z. Gyongyi and H. Garcia-Molina. Web spam taxonomy. In *First International Workshop on Adversarial Information Retrieval on the Web*, 2005.

[20] Z. Gyongyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *Proceedings of the 30th International Conference on Very Large Databases*, pages 576–587, 2004.

[21] A. Heydon and M. Najork. Mercator: A scalable, extensible web crawler. *World Wide Web*, 2(4):219–229, 1999.

[22] B. Huffaker, M. Fomenkov, D. J. Plummer, D. Moore, and K. Claffy. Distance metrics in the internet. In *Proceedings of the International Telecommunications Symposium*, 2002.

[23] M. Koster. A standard for robot exclusion.

http://www.robotstxt.org/wc/norobots.html, 1994.

[24] S. Lawrence and C. L. Giles. Accessibility of information on the web. *Intelligence*, 11(1):32–39, 2000.

[25] H.-T. Lee, D. Leonard, X. Wang, and D. Loguinov. Irlbot: Scaling to 6 billion pages and beyond. In *Proceedings of the 17th International World Wide Web Conference*, pages 427–436, 2008.

[26] M. Najork and J. L. Wiener. Breadth-first crawling yields high-quality pages. In *Proceedings of the 10th International World Wide Web Conference*, pages 114–118, 2001.

[27] C. Olston and S. Pandey. Recrawl scheduling based on information longevity. In *Proceedings of the 17th International World Wide Web Conference*, pages 437–446, 2008.

[28] G. Pant, S. Bradshaw, and F. Menczer. Search engine-crawler symbiosis: Adapting to community interests. In *7th European Conference on Research and Advanced Technology for Digital Libraries*, pages 221–232, 2003.

[29] G. Pant, P. Srinivasan, and F. Menczer. Crawling the web. In *Web Dynamics: Adapting to Change in Content, Size, Topology and Use, edited by M. Levene and A. Poulovassilis*, pages 153–178, 2004.

[30] V. Shkapenyuk and T. Suel. Design and implementation of a high-performance distributed web crawler. In *Proceedings of the 18th International Conference on Data Engineering*, pages 357–368, 2002.

[31] S. Teng, Q. Lu, M. Eichstaedt, D. Ford, and T. Lehman. Collaborative web crawling: Information gathering/processing over internet. In *32nd Hawaii International Conference on System Sciences*, 1999.

[32] P. Yalagandula, S. Nath, H. Yu, P. B. Gibbons, and S. Seshan. Beyond availability: Towards a deeper understanding of machine failure characteristics in large distributed systems. In *Proceedings of the 1st Workshop on Real Large Distributed Systems*, 2004.

[33] D. Zeinalipour-Yazti and M. D. Dikaiakos. Design and implementation of a distributed crawler and filtering processor. In *Proceedings of the Next Generation Information Technologies and Systems*, pages 58–74, 2002.