Web Crawling Using Location Aware Technique

Praveen Kantha\(^1\) Amit Kumar\(^2\)
\(^1\)Assistant Professor \(^2\)Research Scholar
CSE Department
BRCM College of Engineering & Technology, Bahal, Bhiwani (Haryana)

Abstract— Most of the modern search engines are based in the well-known web crawling model. A centralized crawler or a farm of parallel crawlers is dedicated to the process of downloading the web pages and updating the database. However, the model fails to keep pace with the size of the web, and the frequency of changes in the web documents. For this reason, there have been some recent proposals for distributed web crawling, trying to alleviate the bottlenecks in the search engines and scale with the web. In this work, facilitating the flexibility and extensibility of Mobile Agent, we consider adding location awareness to distributed web crawling, so that each web page is crawled from the web crawler logically most near to it, the crawler that could download it the faster. We evaluate the location aware approach and show that it can reduce the time demanded for downloading to one order of magnitude.

I. INTRODUCTION

Internet is considered from many scientists the great revolution of the last decade. The most now application running over the Internet, World Wide Web, managed to get into every scientific laboratory and every school of the modern world, and become one of the most essential tools for knowledge, teaching, researching, even commerce. World Wide Web is expected to contain several billions of available documents, and from many researchers is seen as a big (the biggest) database system. A web crawler, or most usually, a farm of web crawlers are assigned the task of downloading pages from the web, processing them, and integrating the processed results in the search engine database. Web crawling however is limited from several severe bottlenecks. For this reason we recently proposed a distributed web crawling approach able to solve these bottlenecks and scale gracefully with the Internet. Mobile agents who were migrating in the available hosts and performing web crawling of the pages assigned to them in low use hours. More awareness specifically, here we enhanced we propose the delegation of each URL to its most near available web crawler in order to reduce the required time to download the web pages. We suggest two novel ways to enable the location-based delegation, which runs with negligible network and processing overhead and have near-to-optimal results location.

II. RELATED WORK

Indexing the Web has become a challenge due to the Web’s growing and dynamic nature.

A study released in late 2000 reveals that the static and publicly available Web(also mentioned as surface web) exceeds 2.5 billion\(^1\) documents, while the deep Web (dynamically generated documents, Intranet pages, web-connected databases etc.) is estimated to be three orders of magnitude larger \(^2\).

Fig.1: Architecture of Web Crawler

Furthermore, Google \(^3\) now reports more than 3.3 billion documents crawled. Other studies show that the Web is growing and changing rapidly (more than 40% of the pages per week) \(^5\), while no search engine succeeds coverage of more than 16% of the estimated Web size \(^4\). Web crawling (or traditional crawling) has been the dominant practice for Web indexing by popular search engines and research organizations since 1993, but despite the vast computational and network resources thrown into it, traditional crawling is no longer able to catch up with the dynamic Web. Consequently, popular search engines require up to 6 weeks in order completing a crawling cycle for the non-paying sites \(^6\). Moreover, the centralized gathering nature of traditional crawlers and the lack of cooperation between major commercial search engines are two more reasons toward these inadequacies.

The absence of a scalable crawling method triggered some significant research in the past few years. For example, Focused Crawling \(^5\) was proposed as an alternative method but it did not introduce any architectural innovations since it relied on the same centralized practices of traditional crawling. As a first attempt to improve the centralized nature of traditional crawling, a number of distributed
III. LOCATION AWARENESS
Web crawling is a well-studied subject in the research community. However, while proposed in the past that each site should be crawled from the most near crawler [8], there is not much work published about the distribution of the web crawling task to geographically spread locations. The current trend in web crawling involves a farm of crawlers working in parallel and using a high-bandwidth connection to the Internet. This way, all the crawlers are located in the same geographical area, and they all use the same network link to connect to the Internet. Subsequently, this introduces very difficult network and processing bottlenecks in the search engine companies, which often requires very expensive solutions.

Realizing these problems, we recently suggested a geographically distributed web crawling method, with the use of migrating crawlers. Mobile agent’s paradigm was performing the web crawling task in a completely distributed manner. The migrating crawlers were deployed to various nodes in the Internet, which belonged to friendly universities and affiliated companies. All the nodes were network-independent of each other (they were not sharing network resources), and geographically distributed. Location aware web crawling is distributed web crawling enhanced with a mechanism to facilitate the delegation of the web pages to the distributed crawlers so that each page is crawled from the nearest crawler (i.e. the crawler that would download the page the fastest). Nearness and locality are always in terms of network distance (latency) and not in terms of physical (geographical) distance. The distinction between the two types of distances (network and geographical distance) is important, since they are not always analogous. Mobile Crawler enables geographical distribution, location aware web crawling appears to be very important for optimization purposes and for reducing the network overhead that occurs during the crawling.

IV. THE NEED FOR LOCATION AWARENESS
Traditional web crawling facilitates standard Internet resources to run and uses the HTTP/GET and conditional HTTP/GET commands to get the information from the web pages. Since these commands, and their results, are routed through the standard Internet infrastructure, they are routed via the standard routers available in the Internet.

Depended on the distance between the source (web-crawler) and the target (web-server) of the HTTP/GET request, and the structure of the underlying network (intermediary routers and connections), a simple HTTP/GET command can pass through a number of routers until it reaches its final destination. The command according to TCP/IP will be broken in network packets, and each packet will autonomously be routed to the target address by the intermediary routers. At the target (web-server) the packets will be composed and form the original command. The same occurs with the results that normally follow the HTTP/GET command. This process however introduces a certain load in the intermediary routers, which are responsible for sending the packets from one router to another, until the packets reach the destination IP address. To make things worse, web crawlers issue a huge number of such HTTP/GET requests, and thus, introduce an important workload in the intermediary routers, and in the Internet backbone.

V. EVALUATION OF LOCATION AWARE WEB CRAWLING
In order to evaluate location aware web crawling we should compare it with another distributed web crawling system. Since the original Crawler proposal was not sharing the exact same purpose with distributed crawling (with Distributed crawler’s migrating crawlers we were trying to crawl only the local web documents -in the host LAN), a server administrator hosting a web crawler is not necessarily registering a list of web sites for the migrating crawler to crawl. Instead, the migrating crawler receives the web sites to be crawled, apart from the local ones (if the server administrator registers any), from the coordinator. The coordinator, always respecting the preferences set from the host administrator (time-span, maximum size of crawled data etc.) randomly assigns all the URLs from the search engine’s database to the migrating crawlers.

A. The need for probing
The logical distance between two Internet nodes will represent the latency between the nodes (the time that takes for a packet to arrive from the sender node to the receiver
node). We call the task of finding the logical distance between two nodes probing. Probing would ideally be done with the HTTP/GET function, since our target is to crawl each site from the crawler that would perform the downloading (HTTP/GET function) faster. However, this would demand too much time and have a big network overhead, since the crawlers would have to completely download a web page for probing. In this work we use two alternative metrics for probing, which have results with accuracy very close to the accuracy given from the HTTP/GET command, but with significantly lower network and time overhead: (a) the round-trip time between the two nodes, which we can easily measure with the invocation of a ping command from the operating system, and, (b) the time required for the execution of an HTTP/HEAD request. We found these two metrics to be well-suited for representing the logical distance between two Internet nodes (the migrating crawler that has the lowest logical distance for a site - calculated with ping or by timing HTTP/HEAD - is expected to be the one that can download that site faster from all the available crawlers).

Since the migrating crawlers are forced to download web documents from remote machines (not resided in the LAN of the host computer). Nevertheless, this is the case in any proposed distributed web crawling approach that tries to crawl the whole web. In fact, our own version is even better than most of the others, since it performs local processing and compression before transmission of the data back to the database, and manages to reduce the size of the transmitted data by one order of magnitude.

VI. METHOD OPTIMIZATION

While the method can ensure an optimal delegation, it creates an important number of probing messages (ICMP pings, HTTP/HEAD requests, or whatever we choose to use for probing). We can avoid the need for probing each URL from all the agents with various optimizations. First of all, since multiple web sites, can be hosted in the same web-server, we can avoid some probing by checking if the same IP address of the web server hosting the URL is already probed in the past. For this reason, we store the probing results in our database. The probing results do not remain perfectly correct during time, but they do not change often either. More specifically, the probing results do not change unless the underlying route that the network packets follow changes drastically, which does not happen very often. Furthermore, we can detect any important change in the probing results, during crawling of the web pages, by timing each download from the web server. In a case we detect such an important change during downloading, we can return the URL back to the coordinator in order to update the database and re-delegate the URL.

Moreover, targeting to less probes, we can ask from the crawlers to probe the new URL sequentially and not in parallel and stop when we get a probing result less than a threshold. Thus, we will not be able to guarantee the best delegation for all the URLs, but, we will have a sufficient (less than the threshold) delegation, and at some of the times, with much less probes than the original brute force method.

VII. CONCLUSIONS

In this paper we introduced the meaning of distributed location aware web crawling. Our experiments with location aware web crawling revealed that this approach can importantly reduce the required downloading time, thus, optimize our distributed crawling approach and reduce the time spent in the downloading function by one order of magnitude. We also tested an implementation of the location aware web crawling which was using a probing function to estimate the distance between two Internet nodes (in order to find the nearest web crawler for each URL), and had very promising results. However, while improving the crawling throughput importantly, distributed location aware web crawling required an extensive number of probes, resulting to a significant network overhead. Thus, we now need to find other location aware web crawling algorithms that have fewer networks overhead.

VIII. REFERENCES