Distributed Indexing of the Web Using Migrating Crawlers

Odysseas Papapetrou
Computer Science Department
University of Cyprus
cs98po1@ucy.ac.cy

Stavros Papastavrou
Computer Science Department
University of Cyprus
stavrosp@ucy.ac.cy

George Samaras
Computer Science Department
University of Cyprus
cssamara@ucy.ac.cy

1. INTRODUCTION

Indexing the Web has become a challenge due to its growing and dynamic nature. The directly accessible Web (also mentioned as surface Web) exceeds 2.5 billion documents while the indirect Web (dynamically generated documents) is almost three orders of magnitude larger [8]. Moreover, the Web is changing 40 per cent of its contents every month [6] while no search engine succeeds coverage of more than 16% of the estimated web size [7].

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 cannot effectively catch up with the dynamic Web. More specifically, the traditional crawling model fails for the following reasons:

- The task of processing the crawled data introduces a vast processing bottleneck at the search engine site.
- The attempt to download thousands of documents per second creates a network and a DNS lookup bottleneck.
- Documents are usually downloaded by the crawlers uncompressed increasing in this way the network bottleneck. In general, compression is not under full facilitation since it is independent from the crawling task and cannot be forced by the crawlers. In addition, crawlers download the entire contents of a document, including useless information such as scripting code and comments, which are rarely necessary for document indexing.

The absence of a scalable crawling method triggered some significant research in the past few years. Focused Crawling [2], was proposed as an alternative method but 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 methods have been proposed (Harvest [1], Grub [5]).

In this ongoing work, we introduce UCYMicra; a crawling system that utilizes concepts similar to those found in Mobile and distributed Crawling introduced in [3, 4]. UCYMicra extends these concepts and introduces new ones in order to build a more efficient model for distributed Web crawling, capable of keeping up with Web document changes in real time. UCYMicra proposes a complete distributed crawling strategy by utilizing Mobile Agents technology. The goals are (a) to minimize network utilization (b) to keep up with document changes by performing on-site monitoring, (c) to avoid unnecessary overloading of the Web servers by employing time realization, and (d) to be upgradeable at run time.

2. THE UCYMICRA CRAWLING SYSTEM

The driving force behind UCYMicra is the utilization of mobile agents that migrate from the search engine to the Web servers, and remain there to crawl, process, and monitor Web documents for updates. Since UCYMicra requires that a specific mobile agents platform be running at the Web Server to be crawled, it is currently running under a distributed voluntary academic environment spanning in a number of continents.

UCYMicra (Figure 1) consists of three subsystems, (a) the *Coordinator Subsystem*, (b) the *Mobile Agents Subsystem*, and (c) a *Public Search Engine* that executes user queries on the database maintained by the Coordinator subsystem.

The Coordinator Subsystem resides at the Search Engine site and is responsible of (a) maintaining the search database, (b) providing online registration for new Web sites to participate in UCYMicra, and (c) administering the *Mobile Agents Subsystem*. The Mobile Agents Subsystem is responsible for crawling the Web and consists of two categories of mobile agents, namely the *Migrating Crawlers* (or Mobile Crawlers) and the *Data Carries*. Figure 2 shows UCYMicra at work.

As mentioned above, the core of the UCYMicra crawling system are the Java-based Migrating Crawlers. Powered by their inherent mobile capabilities, the Migrating Crawlers can perform the following tasks:

(a) Be dispatched to a newly registered Web server that will participate in UCYMicra.

(b) Crawling: A Migrating Crawler can perform a complete local crawling (either though HTTP or the file system).

(c) Processing: Crawled documents are stripped down into keywords, and keywords are ranked based on their visual properties (font and color), position and occurrence frequency, in order to locally create a keyword index of the web server contents.

(d) Compression: The index of the Web server contents is locally compressed to minimize transmission time between the Migrating Crawler and the Coordinator subsystem.

(e) Data transmission: The compressed index is transmitted to the Coordinator subsystem by the Data Carriers. There, it is uncompressed and integrated into the search database. The choice of using mobile agents for data transmission over other network APIs (such as RMI, CORBA or sockets) is the utilization of their asynchronisity, flexibility and intelligence in order to ensure the seamless transmission of the data.

(f) Monitoring: The Migrating Crawler can detect changes on the Web server contents. Detected changes are instantly processed, compressed and transmitted to the Coordinator subsystem.

(g) Real time upgrades: New code for performing any of the above tasks can be easily deployed since UCYMicra's crawling architecture is based on Java.

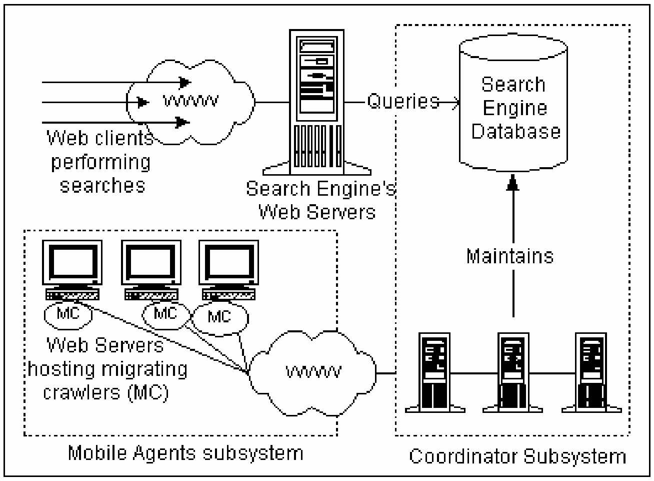
WWW
Web clients performing searches
Queries
Search Engine Database
Search Engine's Web Servers
Maintains
MC
MC
MC
Web Servers hosting migrating crawlers (MC)
WWW
Mobile Agents subsystem
Coordinator Subsystem

Figure 1. UCYMicra's Architecture.

3. EVALUATING UCYMICRA

We compare the performance of the UCYMicra crawling system against traditional crawling in terms of (a) size of data transmitted across the Internet, and (b) total time required for the complete crawling of a given set of documents. For the time being, we study only those two obvious metrics and do not experiment with parameters such as document change frequency. Since it was not feasible to include commercial Web servers in our experiments, we have employed a set of ten Web servers within our voluntary distributed academic environment that span in several continents. Each Web server was hosting an average of 200 Web documents with an average document size of 25 Kbytes. The previous numbers yield 46.2 Mbytes of document data to be crawled by both the traditional crawling and the UCYMicra approach. Due to space limitations, we present our finding for the size of data moved (our findings for the total time required are analogous).

Table 1. Performance Results

Methodology	**Data Moved**
Traditional Crawling	46.9 MB
UCYMicra - no processing, no compression	48.1 MB
UCYMicra - w/processing, no compression	13.3 MB
UCYMicra - no processing, w/compression	8.1 MB
UCYMicra - (w/processing and compression)	2.6 MB

The performance results (Table 1) showed that UCYMicra (row 5) outperforms traditional crawling (row 1) by generating approximately 20 times less data. This is because the Migrating Crawlers *process and compress* the Web documents locally to the Web server. In this way, only the compressed ranked keyword index of the Web server contents is transmitted to the Coordinator subsystem. In the traditional crawling approach, the complete contents of a Web server have to be downloaded by the crawler for centralized processing. In addition, a traditional crawler may request but cannot force a Web server to compress its contents before download.

To get a better insight on our performance results, we performed three more experiments, this time by modifying the UCYMicra approach to perform (or not) local processing or compression. Our results showed that with either processing or compression, the performance gains still hold to some extend. With neither processing nor compression enabled, however, those gains are eliminated since UCYMicra emulates traditional crawling.

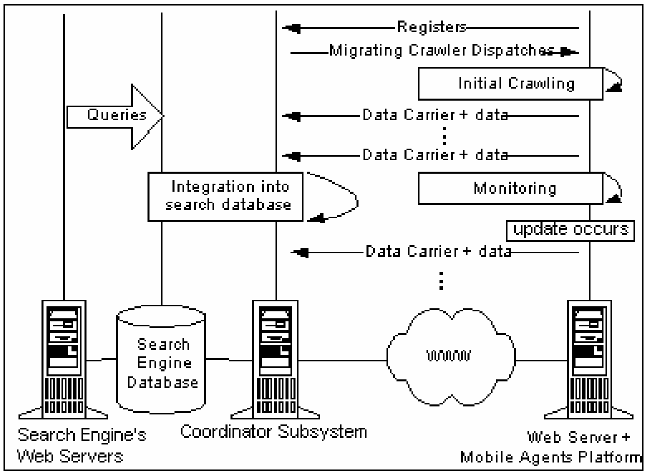

Figure 2. UCYMicra at work.

4. ONGOING WORK

Our ongoing work focuses in extending UCYMicra to support a hybrid crawling mechanism that borrows technology from both the traditional and the fully distributed crawling system. Such a hybrid crawling system will support a hierarchical management structure that considers network locality. Efficient algorithms for work delegation, administration, and result integration are currently under development.

5. REFERENCES

[1] C. M. Brown, B. B. Danzig, D. Hardy, U. Manber, and M. F. Schwartz. The harvest information discovery and access system. In *WWW2*, Chicago, October 1994.

[2] S. Chakrabarti, M. van den Berg, B. Dom. Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery. *WWW8 / Computer Networks* 31(11-16): 1623-1640 (1999).

[3] J. Fiedler and J. Hammer. Using the Web Efficiently: Mobile Crawling. In *Proc. of the 7th Int'l Conf. of the Association of Management (AoM/IAoM) on Computer Science*, 324-329. San Diego, CA, August 1999.

[4] J. Fiedler and J. Hammer. Using Mobile Crawlers to Search the Web Efficiently. International Journal of Computer and Information Science, pp. 36-58, 2000.

[5] Grub: Distributed Internet Crawler. Available at www.grub.org

[6] B. Kahle. Achieving the Internet. *Scientific American*, 1996.

[7] S. Lawrence, C. Lee Giles. Accessibility of information on the web. *Nature*, 400(6740):107 –109, July 1999.

[8] P. Lyman, H. Varian, J. Dunn, A. Strygin, and K. Swearingen. How much information? Available at http://www.sims.berkeley.edu/how-much-info.