MEASURING THAI WEB USING NONTRISPIDER

Surasak Sanguanpong, Punpiti Piamsa-nga, Yuen Poovarawan, Suthiphol Warangrit

Department of Computer Engineering, Faculty of Engineering, Kasetsart University,
Bangkok 10900, Thailand
E-Mail: {nguan, pp, yuen, g4165234}@ku.ac.th

ABSTRACT

This paper presents quantitative measurements and analyses of various issues related to web servers and web documents in Thailand. We developed a web spider to automatically collect the web information and describe the design and its performance. Using the web spider, we found that there are over 700,000 documents and over 8000 web servers on Thai web (information located on the servers registered under `.th` domain.) Then, we present statistical analyses of the current status of Thai web based on information in March 2000.

INTRODUCTION

This paper has two primary proposes. Firstly, it describes a multi-process web spider, called *NontriSpider,* developed as a part of *NontriSearch* search engine [Sanguanpong98]. Secondly, this paper presents quantitative measurements and analysis concerning of web servers in Thailand. Analysis in this paper was based on database crawling on March 20, 2000. We presents several characteristics of Thai web space and investigate certain properties of documents, such as distribution of file sizes, top ten big domains, number and type of embedded images, language usage, etc. Full details of facts and figures are available at http://anreg.cpe.ku.ac.th/links/thaiwebstat/index.html

In the next Section, we describe the design of our NontriSpider and present its performance measurements. The following Section presents the analyses of Thai web. Finally, we conclude the paper and discuss on future work.

WEB SPIDER

Web *spider* (*crawler* or *robot*) is one of the key components of search engine. The main task of spider is to automatically gather HTML documents or web pages from public web servers. A number of spiders will be sent through the whole web to read and analyze all web pages in order to build an index for search engine services. Besides search-engine applications, collection of web pages is also very important for other applications, such as resource discovery, data mining, statistics, etc.

Research and development of web spider has its milestone less than a decade [Eichmann94, Pinkerton94]. Currently, there are well over two hundred of spiders available on the web [Koster93a]. Simple spider runs on a single machine with single-process approach; on the other hand, advanced spiders adopt parallel-and-distributed approach in order to speed-up the data collection. However, there are many cautions in designing spider. For example, the well-behavior spider should not “attack” target web servers by sending too many requests in a short period of time. Other cautions can be found in [Koster93b].

A Design of NontriSpider

NontriSpider is a web spider, which utilized multiple of web collectors. Web collector is a software process to collect information from a targeted web server. The design concept of multiple-process of spider is somewhat straightforward and could be found in [daSilva98, Roehrl99]. Figure 1 illustrates the structure of NontriSpider. The system composes of a "scheduler" and several "collectors." The collectors are assigned to "crawl" the web to gather information by reading and analyzing all HTML pages on an assigned web server. Since data dependencies among collectors are relatively low, the collector is very possible to run in parallel. Therefore, a number of collectors can gather information from different targeted web server simultaneously. For example, while a collector is waiting for an answer from a targeted web server, others collectors can send the request to other servers. Therefore, using multiple collectors improves performance of the spider.

All collectors are managed and controlled by a scheduler. The scheduler is assigned to 1) give each collector a list of targeted URLs; 2) receive all extracted URLs from each collector, and 3) create index key to the URLs. As seen in the Figure, only one scheduler is assigned to manage a whole set of collectors.

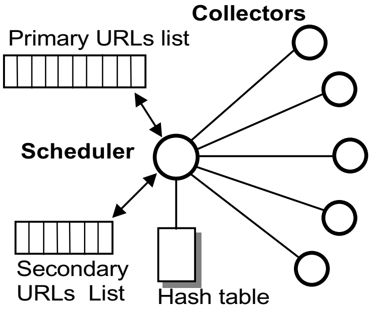

Figure 1:Architecture of NontriSpider

The scheduler has two URL lists: primary and secondary. Primary list contains a list of targeted URL. The scheduler will pick a URL randomly from the primary list and send the targeted address to a collector in order to get information from that address. This "randomly pick" policy is exploited to avoid generating too much sustained requests to a certain server.

The secondary URL list is used to keep addresses, which cannot be accessed successfully. There are many problems of data accesses such as servers are out of services, gateway timeout, or documents are no longer exists, etc. All these problems require different attentions. For example, if an address has a problem of no response or timeouts of servers, the schedule will consider that those servers are temporarily disable. Then, all URL addresses on the primary list that links to those servers will be moved from primary list to secondary list with a flag mark. This technique will prevent the collectors from being stuck in the timeout situations and improve collection rate.

Because many HTML documents could have links pointing to the same documents. Before the scheduler launches any URLs to the collectors, crosscheck verification with other addresses in the primary list is required in order to avoid unnecessary repeated downloads. A hash table of URLs was exploited to store unique URLs for prevent a revisiting of downloaded documents. Algorithm 1 illustrates the scheduler algorithm and Algorithm 2 illustrates the collector algorithm.

Algorithm 1: Scheduler algorithm

```
Scheduler ()
Begin
   init $URLlist  //Initial URLs
   init $HashTable //URLs has table
   loop until empty ($URLlist) and (every idle collector)
   begin
      //Dispatch a URL to a collector
      loop until empty ($URLlist)
      begin
         $URL = getlist ($URLlist);
         $Collector = Find ($ProcessTable, IDLE);
         sendMessage ($Collector, $URL);
      end loop
      // Receive URLlist from collectors
      loop until no message arrival
      begin
         $mesg=receiveMessage ($Collector);
         if ($mesg == IDLE)
            set ($ProcessTable, Collector, BUSY)
         else
            if ($mesg == HYPERLINK)
           begin
              $URL = $mesg
              if not ($URL in $HashTable)
              begin
                 putlist ($URLlist, $URL)
                add ($HashTable, $URL)
             end if
           end if
         end if
      end loop
   end loop
end
```

Algorithm 2: Collector algorithm

```
Collector ()
Begin
   sendMessage (Scheduler, IDLE)
   loop until receive terminated signal
   Begin
      $URL = receiveMessage (Scheduler)
      HTTPget ($URL)
      $URLQueue = filter(parser(URLLink ($URL))
      loop until empty ($URLQueue)
      begin
         getlist ($URL)
         SendMessage (Scheduler, $URL)
      End loop
      sendMessage (scheduler, IDLE)
   end loop
End
```

Performance of Web Spider

NontriSpider is a C/C++ application running on a single 500-MHz Pentium III Xeon/Linux PC, which has 512-MB memory, 2 units of 9.1GB of Ultra SCSI-2 hard disks, and a 100-Mbps Ethernet connection to the router. The router is connected via a 2-Mbps line to THAISARN academic network [Koanatrakool94]. All HTTP traffics to our hosts that reside in Thailand are forced to go through the University cache servers under Cache Infrastructure Project. The University cache server utilized 3 nodes Beowulf clusters. Each node is a 500-MHz Pentium III Xeon PC, which has 1-GB memory and three of 9.1-GB UltraSCSI-2 hard disks. Squid [squid] is exploited on nodes as proxy server software.

We have two experiments of exploiting spiders: with the cache server and without cache server. With the cache server, we tested the spider by launching a number of spider processes. The number of collectors ranges from 10 to 300. We ran 20 times of experiments with the same set of URLs in different periods of day. The best five and the worst five results are discarded. Other remaining results are used to compute average data measurement. To get actual performance of the collection without caching environments, we perform another test by running spiders through a 155 Mbps connection to the university network consortium (UNINET), which has shared 8 Mbps bandwidth to Internet.

The result in Figure 2 shows that collection rates via cache growth linearly and beginning to saturate when the number of collectors close to 130. This is because the cache servers could not handle the rapid requests very well. On the other hand, when the collector gets data directly through the Internet without cache, the data collection rate is up to 140 documents per second (200 KB per second).

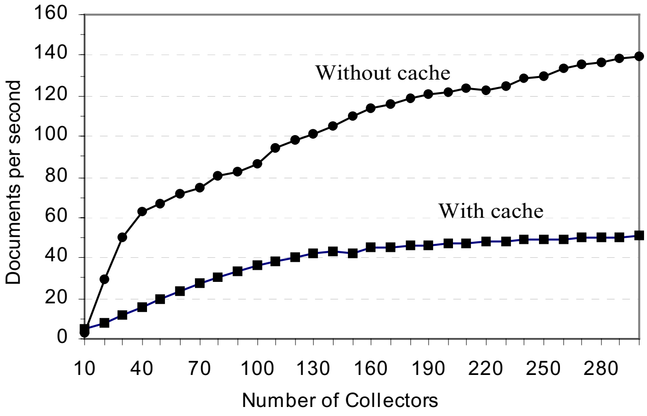

Figure 2: Document Collection by Web Spider

STATISTICAL ANALYSES OF CURRENT THAI WEB

From our knowledge, the only report about web server in Thailand is presented in [Roehrl99] and only summary results is presented without important statistics. In this paper, we collected the data from the Thai web servers and used them to illustrate current status of web information in Thailand, which is information located on the servers registered under `.th` domain. Firstly, we clarify the difference between web servers and machines. Then, following eight Sections are discussions and analysis of experimental results in the eight issues, which are as follows:

1) Size of Thai web
2) Distribution of documents
3) Number of servers and number of documents in 10 largest domains
4) Distribution of page sizes
5) Number of image counts
6) Behavior of outgoing URLs in documents
7) Distribution of File types
8) Languages used in documents

Web servers and physical machines

A web server in this paper is a program that provides HTTP services. It is counted by a unique URL address, not a physical machine. One machine may support multiple web servers simultaneously. To precisely get the number of machines that are used as web servers, we count IP addresses of machines rather than their hostnames.

Size of Thai Web

Five metrics are used to measure the size of Thai web under `.th` domain. The metrics are 1) number of HTML documents; 2) size of HTML pages; 3) number of web-server addresses; 4) number of web-server machines; and 5) number of third-level domains.

Table 1 shows the current measured metrics. We found that there are totally 769,169 HTML documents on 8,520 web servers or 90 HTML documents per server. The total size of HTML documents is around 10 gigabytes. For physical machines, we found there are 4,420 units; therefore, in average, each machine supports two web servers. Also, 2830 of third level domain are discovered (compare to 1931 domain reported by [Roehrl99].

Table 1: Information of web servers under `.th` domain

Type	Quantity
Number of HTML documents	769,169
Total documents size (GB)	7.6
Number of web-server addresses	8,520
Number of web-server machines	4,420
Number of Third-Level Domains	2,830

Distribution of Documents

All web-server addresses in Table 1 are categorized into eight major sub-domains: `ac.th` (2756), `co.th` (4191), `go.th` (500), `in.th` (264), `mi.th` (24), `net.th` (218), and `or.th` (566). All categories of sub-domains are shown in Table 2. (Note that the insignificant domain `amazingthailand.th`, which has only 1 server and 173 HTML documents, is not shown.) Over 80% of web servers are in the commercial (`.co.th`) and the academic (`.ac.th`) domain. Recent survey shows that there are 71995 Internet hosts in Thailand [IIRC00]. Therefore, 12% of Thai Internet hosts are web servers.

Table 2: Server classification by domain name

Domain	Number of Web servers	Number of documents
`ac.th`	2756	382485
`co.th`	4191	174283
`go.th`	500	104134
`in.th`	264	5158
`mi.th`	24	5538
`net.th`	218	33010
`or.th`	566	64388

Number of Servers and Number of Documents

In this Section we use total number of web servers and total number of HTML pages on each domain to rank Thai web domains, which are illustrated in Table 3 and Table 4, respectively. As illustrated in Table 3, the domain `chula.ac.th` (Chulalongkorn University) is on top of the list since it has the most number of web servers. As shown in Table 4, the domain `ku.ac.th` (Kasetsart University) is the biggest web server site, when it is ranked by the number HTML documents.

Table 3: Top-Ten domains ranked by number of web servers

Rank	Domain	Number of servers
1	chula.ac.th	236
2	inet.co.th	210
3	kku.ac.th	163
4	tu.ac.th	149
5	cmu.ac.th	144
6	au.ac.th	132
7	ku.ac.th	125
8	rit.ac.th	124
9	psu.ac.th	115
10	buu.ac.th	107

Table 4: Top-Ten domains ranked by number of documents

Rank	Domain	Number of documents
1	ku.ac.th	67678
2	inet.co.th	37501
3	kku.ac.th	35333
4	chula.ac.th	28226
5	mut.ac.th	25170
6	wnet.net.th	21650
7	psu.ac.th	21359
8	krisdika.go.th	20958
9	swu.ac.th	20021
10	buu.ac.th	19305

Besides information shown in Table 3 and Table 4, the distributions of web servers and HTML documents for all domains are illustrated in Figure 3 and Figure 4, respectively.

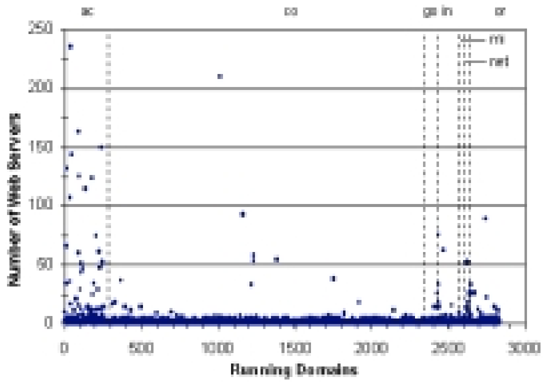

Figure 3: Distribution of web servers

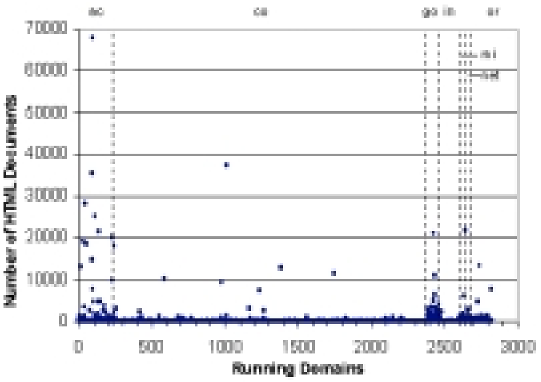

Figure 4: Distribution of HTML documents

Distribution of Document Page Sizes

Figure 5 shows the size distribution of the web pages. Sizes of Over 55% of web pages are between 1 to 8 KB. We show the file size distribution in log scale because the right tail of the distribution is “heavy-tailed”. We can see that majority of pages are small documents. However, variance of distribution is very large (658,640,896 in our experiment). The right tail of the distribution could be fit with the Pareto distribution [Crovella96, Willinger98, Barford99].

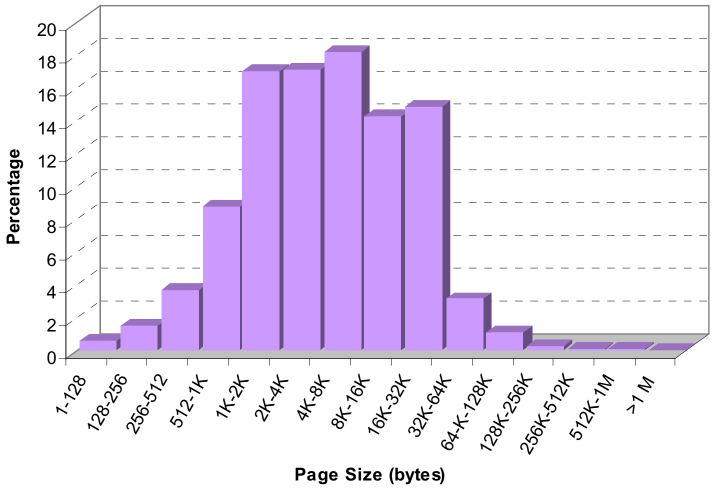

Figure 5: Distribution of page sizes

Number of Image Counts

Figure 6 shows the distribution of embedded image counts. The right tail of the distribution is again "heavy-tailed", similar to the distribution of file size. This Figure shows only pages, which contain 0 to 15 images. We found that over 50% of pages contain 1 to 10 images and 24.67% of pages contain no images. Average number of images per page is 10.

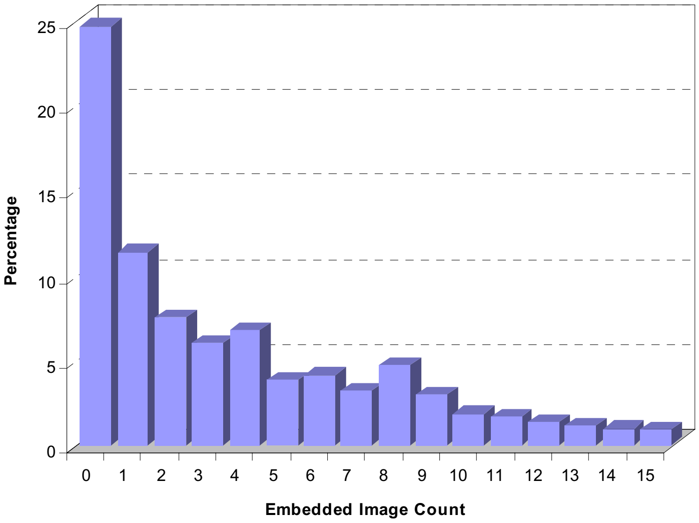

Figure 6: Distribution of image counts

Behavior of Outgoing URLs in documents

Overall characteristics of hyperlinks in Thai web are interesting. Figure 7 shows pages, which has HTML links no more than 30 links. Around 55% of all pages contain 1-10 outgoing links. Around 17% of web pages have no outgoing links. Average number of HTML links in a page is 13 links per page.

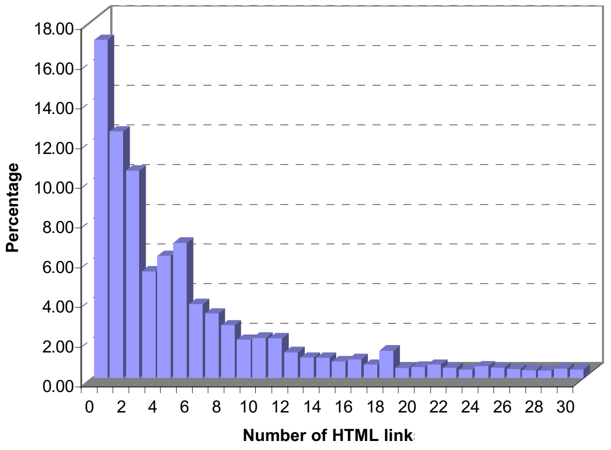

Figure 7: Distribution of Outgoing URL Counts

We analyze the outgoing links into 4 categories: 1) Inbound links point into the same web server; 2) Inbound links point to different web servers; 3) Outbound links point to `.th` servers; and 4) Outbound links point to non-Thai domains. We classify into further each of four categories by number of links in a page and then we found that:

1. For any web pages, which have fewer than 40 links/page, eighty percent of their links falls into the first category.
2. For any web pages, which have more than 40 links/page, fewer than sixty percent of their links are inbound links and the rest are outbound links.
3. By manually random investigation, we also found that most of these web pages contain links to software download from foreign sites.

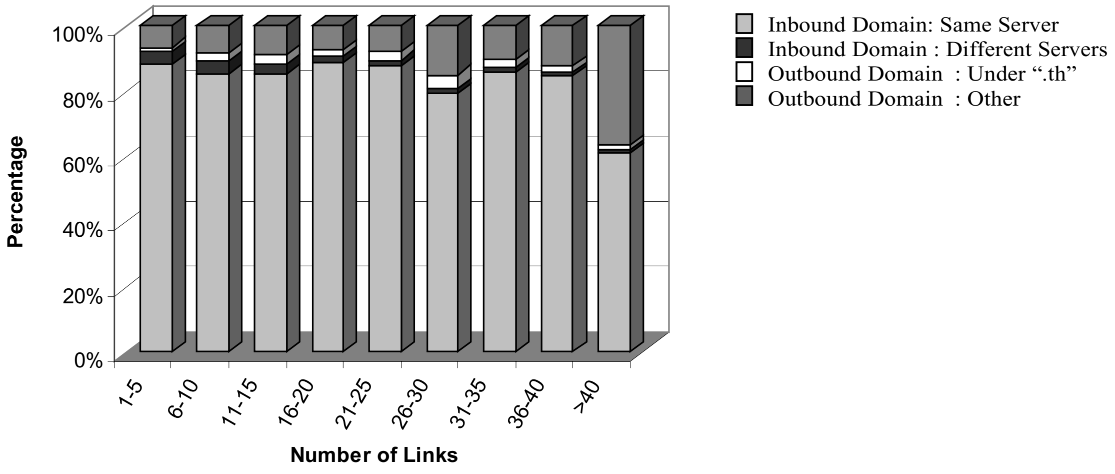

Figure 8: Distribution of Outgoing links

File Types

In this Section, we investigated the file-type distribution in Thai web. Classification has been done by the standard suffix used in file names e.g. .html, .htm, .jpg, .gif, etc. File names without suffix are classified as unknown. We found that seventy percent of all file types are HTML documents. The distribution of all file types is shown in Figure 9.

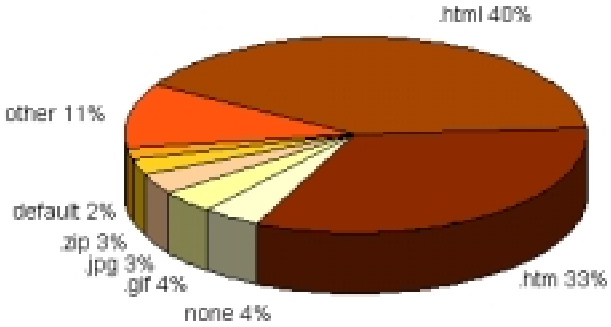

Figure 9: Distribution of File types

Languages Used in Documents

We built a simple language identification based on pattern detection. Our program is simple and it can identify whether a page is written in Thai or not. From Figure 10, it is very surprised that English is major language used in Thai web. Number of pages that are written in English is around 66% of total number of pages. Almost all domains, Thai pages contribute less than 50%; except in `go.th` and `mi.th` domains that have more than 50%. However, as denoted in Table 1, `go.th` and `mi.th` domain contributed only 13.54% and 0.72% of overall web pages respectively.

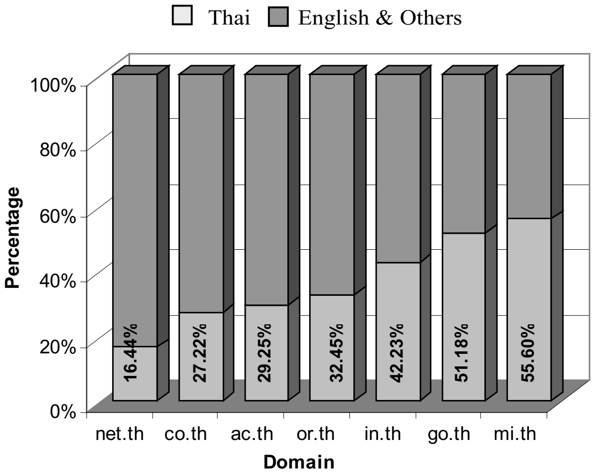

Figure 10: Languages used in each major domain

CONCLUSIONS AND FUTURE WORK

Quantitative measurement and statistics of Thai web are presented and analyzed. The design of Web spider is also introduced.

We plan to collect the Thai web pages every month to track the growth rate. Furthermore, there are many Thai servers that are registered under foreign domain. (e.g. `.com`, `.net`, `.org`, etc.) Recently investigation, there are 761 Thai servers that are registered under foreign domains [Upatisong00]. They would be included in the future measurement.

REFERENCES

[Barford99] P. Barford, A. Bestavros, A. Bradley, and M. Crovella. *Changes in web client access patterns: Characteristics and caching implications* World Wide Web, Special Issue on Characterization and Performance Evaluation, 1999.

[Crovella96] M. Crovella and A. Bestavros. *Self-similarity in World Wide Web traffic: Evidence and possible causes*. ACM SIGMETRICS Conference on Measurement and modeling of Computer System, pages. 16-169, May 1996.

[daSilva98] A. da Silva, E. Veloso, P. Golghe, B. Ribeiro-Neto, A. Laender and N. Ziviani. *CoBWeb — A Crawler for the Brazilian Web.* Proceedings of the String Processing and Information Retrieval Symposium & International Workshop on Groupware, 1998.

[Eichmann94] D. Eichmann. *The RBSE Spider- Balancing Effective Search Against Web Load*, Proceedings of the First International Conference on World Wide Web , pages 113-120, Geneva, Switzerland, May 1994.

[IIRC00] Internet Information Resource Center. *Domain counts and Host counts for .TH domain*. May 2000. <http://ntl.nectec.or.th/internet/domainname/ WEB/>.

[Koanantakool94] T. Koanantakool, T. Tansethi, M. Kulatumyotin, *THAISARN: The Internet of Thailand*, July 1994. <http://ntl.nectec.or.th/thaisarn/thaisarn-body.html>.

[Koster93a] M. Koster. *Database of Web Robots, Overview.* 1993. < http://info.webcrawler.com/mak/projects/robots/active/html/index.html>

[Koster93c] M. Koster. *A Standard for Robot Exclusion*. 1993. <http://info.webcrawler.com/mak/projects/robots/norobots.html>.

[Koster93b] M. Koster. *Guidelines for Robot Writers*. 1993. <http://info.webcrawler.com/mak/projects/robots/guidelines.html>

[Pinkerton94] B. Pinkerton. *Finding What People Want: Experience with the WebCrawler*. Proceedings of the Second International Conference on World Wide Web, October 1994.

[Roehrl99] A. Roehrl., M.Frey, and R. Roehrl. *World Wide Web Robot for Extreme Datamining with Swiss-Tx Supercomputers*. Interim Report IIASA IR-99-20, 1999.

[Sanguanpong98]. S. Sanguanpong and S. Warangrit. *NontriSearch: Search Engine for Campus Network*. National Computer Science and Engineering Conference, Bangkok, Thailand, 1998.

[Upatisong00] V. Upatisong, *Electronics Personal Communications*, February 2000.

[Willinger98] W. Willinger and V. Paxson. *Where mathematics meets the Internet*. Notices of the AMS, 45(8): 961-970, 1998.