# Distributed High-Performance Web Crawler Based on Peer-to-Peer Network

**Chapter** · December 2004

DOI: 10.1007/978-3-540-30501-9_13



**5 authors**, including:

Minglu Li
Shanghai Jiao Tong University
**561** PUBLICATIONS **7,987** CITATIONS

# Distributed High-Performance Web Crawler Based on Peer-to-Peer Network

Liu Fei, Ma Fan-Yuan, Ye Yun-Ming, Li Ming-Lu, and Yu Jia-Di

Department of Computer Science and Engineering,
Shanghai Jiaotong University, Shanghai,
P. R. China 200030
{liufei001, my-fy, ymm, li-ml, yujiad}@sjtu.edu.cn

**Abstract.** Distributing the crawling activity among multiple machines can distribute processing to reduce the analysis of web page. This paper presents the design of a distributed web crawler based on Peer-to-Peer network. The distributed crawler harnesses the excess bandwidth and computing resources of nodes in system to crawl the web. Each crawler is deployed in a computing node of P2P to analyze web page and generate indices. Control node is another node to being in charge of distributing URLs to balance the load of the crawler. Control nodes are organized as P2P network. The crawler nodes managed by the same control node is a group. According to the ID of crawler and average load of the group, crawler can decide whether transmits the URL to control node or hold itself. We present an implementation of the distributed crawler based on Igloo and simulate the environment to evaluate the balancing load on the crawlers and crawl speed.

## 1 Introduction

The architecture of the current crawler [1] [2] is based on a single architecture design. Centralized solutions are known to have problems like link congestion, being single point of failure, and expensive administration. To address the shortcomings of centralized search engines, there have been several proposals [3] to build decentralized search engines over peer-to-peer networks. Peer to Peer system are massively distributed computing systems with each node communicating directly with one another to distribute tasks or exchange information or accomplish task. In our system each crawler is deployed in a computing node to analyze web page and generate indices. Control node is another node to being in charge of distributing URLs to balance the load of the crawler. Control nodes are organized as P2P network---CAN [4]. The crawler nodes managed by the same control node is a group. According to the ID of crawler and average load of the group, crawler can decide whether transmits the URL to control node or hold itself. We present an implementation of the distributed crawler based on Igloo and simulate the environment to evaluate the balancing load on the crawlers and crawl speed.

K.-M. Liew et al. (Eds.): PDCAT 2004, LNCS 3320, pp. 50–53, 2004.

## 2 The Structure of Single Crawler

We have implement Igloo that has three versions. This paper uses crawler in Igloo version 1.2 as single crawler to construct our system. First we introduce the architecture of single crawler (Fig. 1).

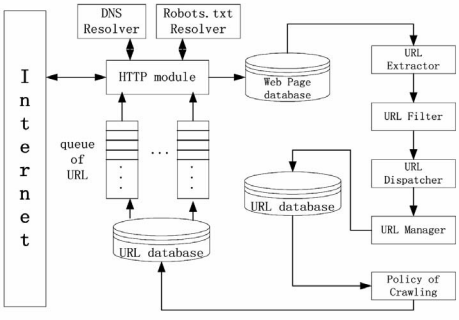



**Fig. 1.** The structure of single crawler

Each crawler can get IP of host with URL by DNS. Then it downloads the web page through HTTP module if Robot allows access to the URL. URL extractor extracts the URL from the downloaded web page and URL filter check whether the URL accord with the restrictions. Then the crawler uses hash function to compute the hash ID of URL. The crawler inserts the URL into its URL database. Policy of crawling is used to sort the rank of pages to make higher important resource to be crawled more prior. We adopt the PageRank [5] method to evaluate the importance of web page. HTTP module consists of several download threads and each thread has a queue of URL.

## 3 Control Node Overlay Network

P2P overlay network is scalable and robust so it fits to organize these control node. We organize these control nodes in CAN. Our design centers around a virtual 4-dimensional Cartesian coordinate space. This coordinate space is completely logical and bears no relation to any physical coordinate system. At any point in time, the entire coordinate space is dynamically partitioned among all the control nodes in the system such that every control node owns it individual, distinct zone within the overall space. We assume that IP of node is the identifier of control node in it. We can regard IP as a point in a virtual 4-dimensional Cartesian coordinate space which is defined as $S_a$*={(0,0,0,0),(255,255,255,255)}*.

## 4 The Architecture of Our System

The ID of crawler is pair *(IP, number)* where *IP* is the IP of the node crawler being in and *number* is a random number that is not used by active crawlers start before the crawler starts. The *number* is one of the values of the crawler. Each crawler that joins

our system must know at least one control node. Then the joining crawler *P1* sends packet containing its *(IP, number)* to the control node *R1* to ask for joining. *R1* checks the IP of *P1*. If the IP of *P1* is in the space controlled by *R1*, *R1* records *(IP, number)* of *P1*. Otherwise *R1* transfer *(IP, number)* to its neighbor which coordinate is closest to the IP of *P1*. Then the neighbor does the same work as *R1* until find a control node which controls the space containing the IP of *P1*. In this way crawlers are organized by control node. The control nodes are used to collect URLs and redistribute them to crawlers to balance the load of crawler.

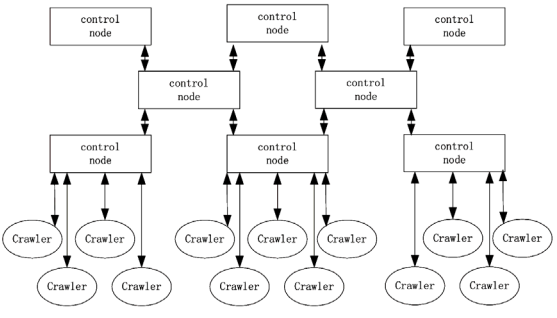
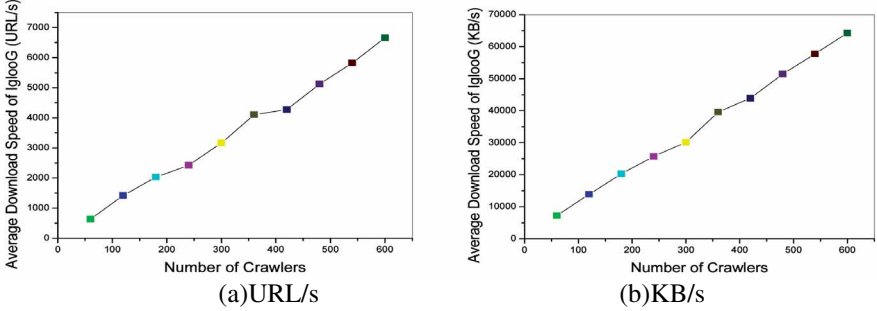



**Fig. 2.** The architecture of our system

Control node computes average load of its group periodiacally and send it to all crawler nodes in its group. If the load of crawler node is lighter than average load, it holds itself. Otherwise it sends the URL to control node to distribute the URL. Fig. 2 shows the architecture of our system.

## 5 Experimental Results

We use GT-ITM models to obtain 7 groups of nodes. One group contains 30 nodes that are used as CAN that organize information services. We implements simulation of CAN. Each of the other 6 groups of nodes contains 100 nodes and every node runs a crawler. We use seven 1.7GHz Pentium IV machine with 1GB of memory as our experiment hardware. One of these computers is used to run CAN that organizes




(a)URL/s (b)KB/s

**Fig. 3.** The average download speed of our system

information services. Each of the other 6 machines run 100 nodes to run crawlers. Fig. 3 shows the average download speed of our system scales linearly as the number of participating crawlers increase. So we know crawlers of our system are load-balanced.

## 6 Conclusions and Future Work

This paper presents the design of a distributed web crawler based on P2P network. This distributed web crawler is based on our previous work Igloo. The control nodes are organized with P2P network. URLs are collected by control node according to the ID of the crawler and the load of the crawler. In this way our system are load-balanced and can scale up to the entire web and has been used to fetch tens of millions of web documents.

**Acknowledgements.** This paper is supported by 973 project (No.2002CB312002) of China, ChinaGrid Program of MOE of China, and grand project of the Science and Technology Commission of Shanghai Municipality (No. 03dz15026, No. 03dz15027 and No. 03dz15028).

## References

1. Sergey Brin, Lawrence Page, Google: The Anatomy of a Large-Scale Hypertextual Web SearchEngine" Proceedings of the 7th International World Wide Web Conference, pages 107-117, April 1998
2. Allan Heydon and Marc Najork, "Mercator: A Scalable, Extensible Web Crawler", *World Wide Web*, 2(4):219–229, 1999
3. J. Li, B. T. Loo, J. Hellerstein, F. Kaashoek, D. Karger, and R. Morrris. On the Feasibility of Peer-to-Peer Web Indexing and Search. In *IPTPS 2003*
4. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable -addressable network. In *ACM SIGCOMM'01*, August 2001
5. Henzinger M R. Hyperlink analysis for the Web. IEEE Internet Computing, 2001, 5(1): 45-50