MULTI AGENT SYSTEM FOR DISTRIBUTED DOWNLOAD

Şova, Ioan[1*] Craus, Mitică[2]

"Gh. Asachi" Technical University
Department of Computer Science and Engineering
700050 Iaşi, Romania

Abstract: In this paper is presented the design and implementation of a distributed download system. The system is based on intelligent agents. It is extensible, robust; it can be also easily integrated in more complex systems. The proposed system yields good performance results compared to other similar systems.

Keywords: multi-agent system, distributed download, Web

INTRODUCTION

In the last decade the volume of information available on the World Wide Web had an exponential growth. As a result, there is an increase in the chance that information needed by the end users is available on the internet, however the problem of information retrieval should be considered. In order to locate the information on Internet, the most popular tools are the search engines. They have a large collection of web pages and these pages are gathered by using web crawlers. A web crawler should have a strategy to traverse the web pages and it should have an optimized mechanism for downloads. The scientific community had explored in the last years different methods of web searching [Arasu et al.]. If for the traverse strategy there is an interest of the scientific community for the second aspect there are few efforts [Shkapenyuk & Suel].

Obviously, the search engines have powerful crawlers, but unfortunately, because of competitive issues, there are few technical details available. One system described in the literature is Mercator that was used for the first time by Altavista search engine [Heydon & Najork]. An initial version of the Google crawler is also described in the literature [Brin & Page]. Unfortunately it is not described in its full details. The Google crawler uses a distributed system for crawling, with multiple machines implied in the process. Internet Archive is another reference point in the web crawling field [Burner].

The main objective of this paper is the design and implementation of a distributed system for resource downloading. This system is a multi agent one that makes use of intelligent agents technology. The system can be integrated in larger and complex systems that require a download module. In the design of the system aspects related to robustness, scalability and extensibility where taken into account.

DISTRIBUTED DOWNLOAD SYSTEM

The download system architecture is presented in Figure 1. The proposed system has a modular structure. It consists of a set of distributed, autonomous agents that cooperate in order to achieve their goal. The execution environment for these agents is provided by PASIBC agent platform [Şova et al.]. The agents have BDI architectures and the communications are encoded by using the KQML language. Special download RDF ontology was built and the agents submit to it. In fact the system is composed from specialized agents that can be distributed on different agent platforms, located on different machines so that better performance to be achieved. A short description of the agents will be provided in the next paragraphs. The requests are launched by using the .NET HTTPWebRequest class.

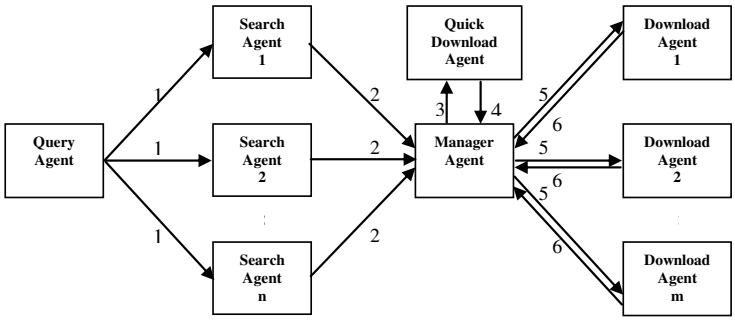

Fig. 1 System architecture

The Query Agent (QA) has the responsibility of keywords reception and to query building. This query will be than sent on Internet. For the effective launch of the query, the QA agent will use a set of search agents (1). The set of search agents is contained in the knowledge base of the QA agent. The keywords can be sent to the QA either by another agent or directly by the user from a GUI interface.

The Search Agent (SA) is the agent specialized in obtaining the results for a specific search engine. The system can have multiple such agents. There can be, depending on the application needs, agents specialized in different search engines or multiple agents specialized in the same search engine. The SA agent will send, alongside with the used keywords, the obtained URL list to its Manager Agent (2).

The Manager Agent (MA) is the pivotal agent of the system and it maintains multiple data structures for download management. It will maintains, for example, a "must be downloaded URL registry" that is modeled by a standard FIFO queue. The MA agent receives from the search agents XML formatted URL lists. It will filter the duplicates so that the already downloaded URLs will be eliminated. In order to obtain more detailed information (like file size, last modified date) about a specific URL, the MA agent has the ability to launch request to Quick Download Agent (3). This

information can be used in the process of selecting the download agent that will process the URL (5). The knowledge base of the MA contains the registered agents of QDA or DA types.

The Quick Download Agent (QDA) is a subtype of download agent specialized in fulfilling of a frame with detailed information for an URL. QDA receives from the MA agent an URL (3) and will returns to this the set of information related to the URL (4). According with the necessities the system can have QDA agents distributed on different platforms.

The Download Agent (DA) represents the agent that effectively performs the download as a result of a request received from the MA agent (5). The agent will inform the MA agent about the result of the download (either success or the error code). In this version the DA agent has a GUI form, so that its status can be monitored. It performs the download in a separate thread, in a synchronous way. In order to obtain better performance results, more DA agents should be used and they must be distributed on different platforms located on different machines.

EXPERIMENTS

The system used in our experiments is composed from four PASIBC platforms distributed on four computers with 1GHz CPU, 1 GB RAM and a Windows XP operating system. On one platform, in order to minimize the communication time, we hosted one manager agent, one query agent and the search agents. The system has three types of search agents specialized for the most popular search engines and directories. Search agents for Google, Altavista and Yahoo were used. We have three agents for each search agent type. The first agent will retrieve the first 10 results, the second agent will retrieve the 11 to 20 results and the last agent will return the 21 to 30 results. Thus a total of nine search agents exist. On the other three platforms we hosted 30 download agents on each of them so that 90 download agents exist in the system.

Launching the above system we achieved an average download rate of 11.12 documents/sec (including search engine requests) and 211.3 KB/sec. The average HTML page was 19 KB; the histogram of the sizes of successfully downloaded documents is presented in Figure 2. The comparison with results obtained by existing web crawlers has little relevance because they are influenced by the hardware configuration used and that our system is not a web crawler, but a distributed download system.

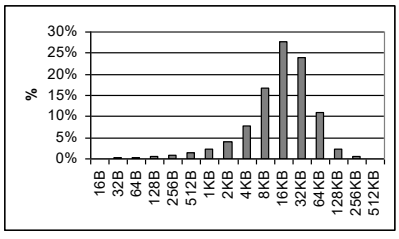

Fig. 2 The histogram of the sizes of successfully downloaded documents

CONCLUSIONS

By building this system we proved that the agents can be used successfully to implement a distributed download system. The proposed system is an extensible one, new specialized agents can be easily added so that the system can benefit form new functionalities. Such functionalities can be indexing or clustering capabilities. Our system can be easily integrated in more complex systems.

The future work will be especially focused to running the system for more days, to monitor the results and to make the eventually necessary adjustments. In order to achieve this, we will need to implement the ability to submit to a maximum download bandwidth so that we will not disrupt the normal university activity. A failure and recovery mechanism will be equally needed.

REFERENCES

[Arasu et al.] Arasu A., Cho J., Garcia-Molina H., and Raghavan S.; [2001] *Searching the web*; ACM TOIT; 1; 1; 2-43;

[Brin & Page] Brin S. and Page L.; 1998; *The anatomy of a large-scale hypertextual Web search engine*; WWW7; Brisbane, Queensland, Australia; 107—117;

[Burner] Burner M.; [1997]; *Crawling towards Eternity: Building an archive of the World Wide Web*; Web Techniques Magazine; 2; 5; 37-40;

[Heydon & Najork] Heydon A., Najork. M.; [1999]; *Mercator: A scalable, extensible web crawler.*; W3J; 2; 4;:219–229,

[Shkapenyuk & Suel] Shkapenyuk V., Suel T.; 2002; *Design and Implementation of a High-Performance Distributed Web Crawler*; ICDE 2002; San Jose, California,USA;

[Şova et al.] Şova I., Leon F., Gâlea D.; 2004; *Improving the Performance of Knowledge Intelligent Agents using a Distributed Framework*; SACCS 2004; Iaşi, Romania; 50