An Approach of Crawlers for Semantic Web Application

José Manuel Pérez Ramírez[1] , Luis Enrique Colmenares Guillen[1]

[1] Benémerita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
BUAP – FCC, Ciudad Universitaria,
Apartado Postal J-32,
Puebla, Pue. México.
{ mankod, lecolme}@gmail.com

Abstract. This paper presents a proposal for a system capable of retrieval information from the processes generated by the system Yacy. The information retrieved will be used in the generation of a knowledge base. This knowledge base may be used in the generation of semantic web applications.

Keywords: Semantic Web, Crawler, Corpora, Knowledgebase.

1 Introduction

A knowledgebase is a special type of database for managing knowledge. It provides the means to collect organize and recover knowledge in a computed way. In general, a knowledgebase is not a static set of information it is a dynamic resource that maybe have the ability to learn. In the future, Internet will be a complete and complex knowledgebase, already known as *semantic web* [1].

Some examples of knowledge base are: a public library, an information database related to a specific subject, *Whatis.com, Wikipedia.org, Google.com, Bing.com and Recaptcha.net.*

Investigate related to ***Generation Automatic of a specialized corpus*** from the Web is present in [2], this investigate have a reviews of methods to process knowledgebase that generates specialized *corpus.*

In section 2 we present related work to *semantic web* in order to comprehend the benefits that may be obtained by elaborating them.

In Section 3 we describe the challenges and we explain the problems that could be have if you tried to use Google Search for getting information or tried to retrieval information of queries to Google.

Section 4 the methodology to use for solving the problem. And section 5, conclusions and ongoing work.

We continue this paper present a form abstract to describe a **Query Processing on the Semantic Web** [8] is as follows Fig. 1

1. A query with a data type.
2. A server that sends queries to the servers decentralized indexing. The content found on the servers is similar to indexing a book index indicates which pages contain the words that match the query.
3. The query travels to the servers where documents stored documents are retrieved are generated to describe each search result.
4. The user receives the results of its semantic search which has already been processed in the semantic web server.

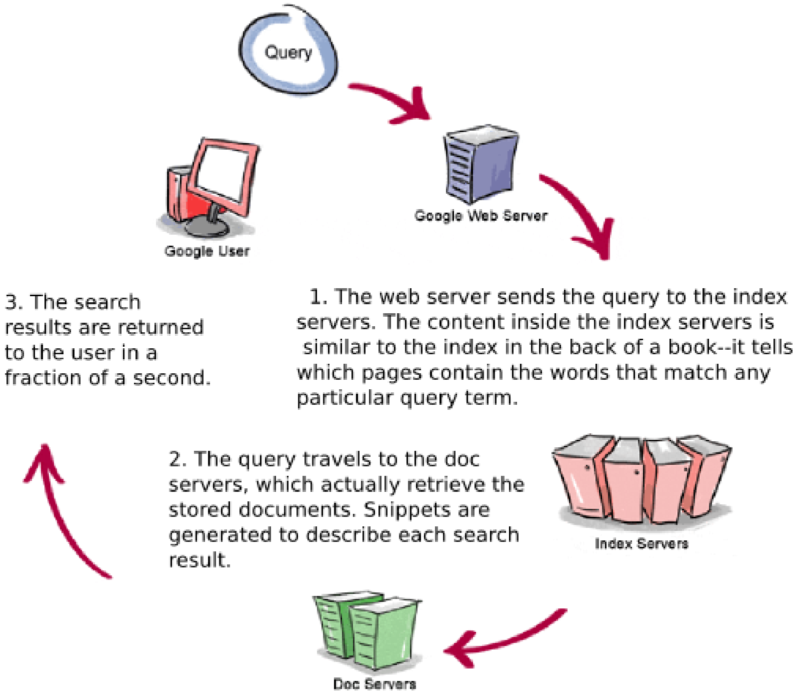

Query
Google Web Server
Google User
3. The search results are returned to the user in a fraction of a second.
1. The web server sends the query to the index servers. The content inside the index servers is similar to the index in the back of a book--it tells which pages contain the words that match any particular query term.
2. The query travels to the doc servers, which actually retrieve the stored documents. Snippets are generated to describe each search result.
Index Servers
Doc Servers

Fig. 1. Querying the Semantic Web.

2 Related Work

Nowadays, the investigation related to ***retrieval information*** on the web has a different result like: knowledgebase, web sites dedicated to retrieval information, Wikipedia, Twine, Evri, Google, Vivísimo, Clusty, etc.

An example of a company that working with "retrieval information" is Google Inc, one of their products is Google Search this *web search engine* is the one of the most-used search engine on the Web [9], Google receives several hundred million queries each day through its various services [10].

This kind of example it's necessary for the following analogy: *For what reason Google doesn't put their information of their knowledgebase under domain public?* And the answer it's very simple: because their *information* or their *knowledgebase it's* ***money.***

In section 3 we explain some form of extract information of Google Search only a protected few of information it's impossible retrieval many information of Google Search whit the idea to generate knowledgebase this because Google protects their information of their queries.

Another kind of knowledgebase are:

2.1 Wikipedia

A specific case is Wikipedia, a project to write a free communitarian encyclopedia in all languages. This project have 514 621 articles today. The quantity and quality of the articles present an excellent knowledgebase for the creation of semantic webs.
We present some ways to obtain semantic information from Wikipedia: from its structure, from the collected notes of the people that contributes and from the existent links in the entries.

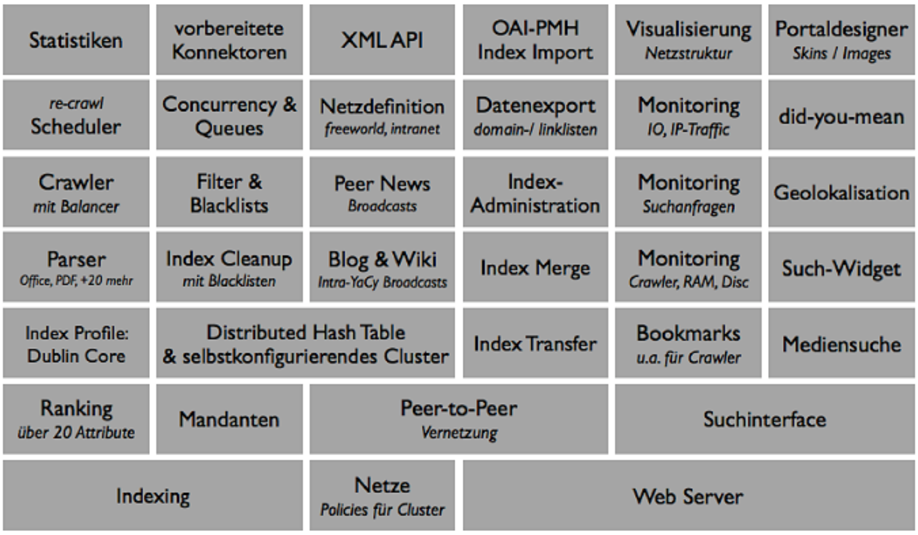
2.2 Twine

Twine is a tool for storage, organizes and shares information, all of it with an intelligence provided by the platform that analyzes the semantic of the information and classifies automatically [7]. The main idea is to save users from labeling and connecting related content and leave this work to Twine, bringing more value and storage the contents next to the information about its meaning.

3 Challenges

The principal challenge is development a system with the capacity of works with Yacy for retrieval information of Indexing Process and generate information this information will be essential for produce knowledgebase.

We present in the figure 5 all modules of yacy, so the module to development will be works with some of these modules.



Statistiken | vorbereitete Konnektoren | XML API | OAI-PMH Index Import | Visualisierung *Netzstruktur* | Portaldesigner *Skins / Images*

re-crawl Scheduler | Concurrency & Queues | Netzdefinition *freeworld, intranet* | Datenexport *domain-/ linklisten* | Monitoring *IO, IP-Traffic* | did-you-mean

Crawler *mit Balancer* | Filter & Blacklists | Peer News *Broadcasts* | Index-Administration | Monitoring *Suchanfragen* | Geolokalisation

Parser *Office, PDF, +20 mehr* | Index Cleanup *mit Blacklisten* | Blog & Wiki *Intra-YaCy Broadcasts* | Index Merge | Monitoring *Crawler, RAM, Disc* | Such-Widget

Index Profile: Dublin Core | Distributed Hash Table & selbstkonfigurierendes Cluster | Index Transfer | Bookmarks *u.a. für Crawler* | Mediensuche

Ranking *über 20 Attribute* | Mandanten | Peer-to-Peer *Vernetzung* | Suchinterface

Indexing | Netze *Policies für Cluster* | Web Server

Figure 5. Components of Yacy

The principal question is:

What we can do to get information under domain public.

It's very simple we use the very popular Wikipedia

Wikipedia is a project of the Wikimedia Foundation. More than 13.7 million of its articles have been drafted in conjunction with volunteers from all over the world and practically every one of them may be edited by any person that may have access to Wikipedia. Actually it is the most popular reference work on the internet.

This project of dynamic content like Wikipedia illustrates the information that have great potential to be exploited.

Otherwise Google Search is one of the most-used search engine provides at least 22 special features beyond the original word-search capability. These include synonyms, weather forecasts, time zones, stock quotes, maps, earthquake data, movie showtimes, airports, home listings, and sports scores.

And maybe you could be thinking:

For what reason the people don't use a Google Search for get all the **knowledgebas**e *about topic specific and this* **knowledgebas**e *could be export to file of text plan with the possibilities of management this and generate corpus.*

Very simple is the answer because the information of Google is their information and gold for company.

It the past Google Inc. allowed the retrieval information from any kind of query[3].

Google allowed the retrieval information based on their form and methods like ***University Research Program for Google Search*** [10] but any kind of answered we get of this project when we make the inscription to this program.

Another way to exploit Google Search knowledge is using scripts, APIS [3], programming languages such as AWK, development tools like SED or GREP, all of them analyzed in [2] but with few results and we need a lot of information for create *knowledgebase*.

3.1 Considerations

1. Create a module with the goal to connect this with YACY and retrieval information of their crawlers.
2. Export a set of information related with a topic in plain text.
3. Management information of web site like Wikipedia.org.
4. Index the content of this kind of retrieval information in storage local.
5. Public the module in the web and share the *knowledgebase.*

4 Methodology

This section gives a description of the project taking into consideration the design that will be used to give a solution to the problem of creating the module.

4.1 Project description

The obtained results of the module that connected with Yacy will be used to create semantic webs, corpus and any other project that needs information in a plain text about web content.

Described below are a series of procedures to follow that use as a methodology to implement within the project.

A) Check the modules of Yacy

B) Check the logistic and architecture of Yacy

C) Check the form that Yacy create their crawlers

D) Think in a form of create the Module capable of manage the information of the crawler and generate *knowledgebase*

E) Some of the polices described above are implemented in YaCy [6], the variant to use is the implementation of the JXTA[5] tool and the URI and RDF policies that allow to structure and outline the results, to finally present then in a *semantic* way or *knowledgebase*.

4.2 Development platform

This work is done with YaCY, which is a free distribution search engine, based on the principles of the peer to peer (P2P). Its core is a program written in Java that it's distributed in hundreds of computers, from September 2006. It's called YaCy-peer. Each YaCy-peer is an independent crawler that navigates trough the Internet, and analyzes and indexes web pages found. To storages the indexation results in a common database (called index) which is shared with other YaCy-peers using the principles of the P2P networks [4].

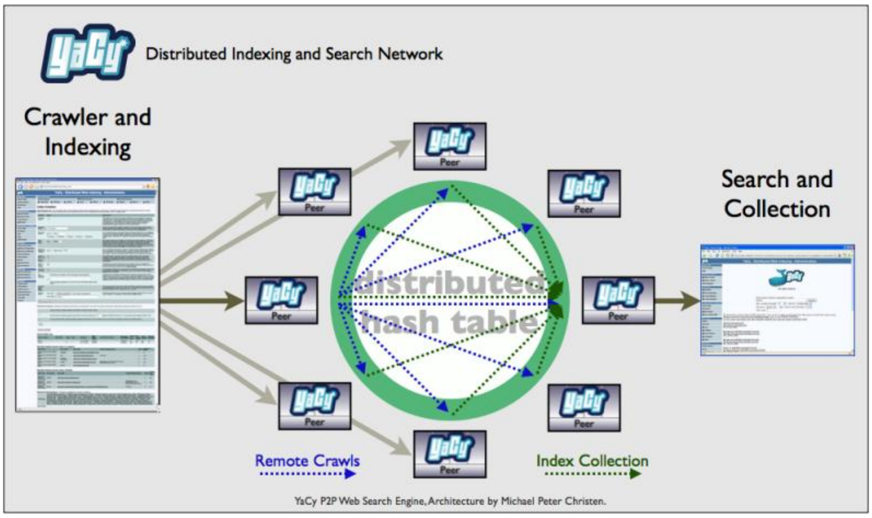

Distributed Indexing and Search Network
Crawler and Indexing
Search and Collection
distributed hash table
Peer
Remote Crawls
Index Collection
YaCy P2P Web Search Engine, Architecture by Michael Peter Christen.

Fig. 2 Distributed indexing process

Compared to semi-distributed search engines, the YaCy-network has a decentralized architecture. All of the YaCy-peers are equal and there is no central server. It may be executed in Crawling mode or as a local proxy server. The figure 2 shows a diagram that describes the distributed process of indexation and the search in the network for the YaCy crawler.

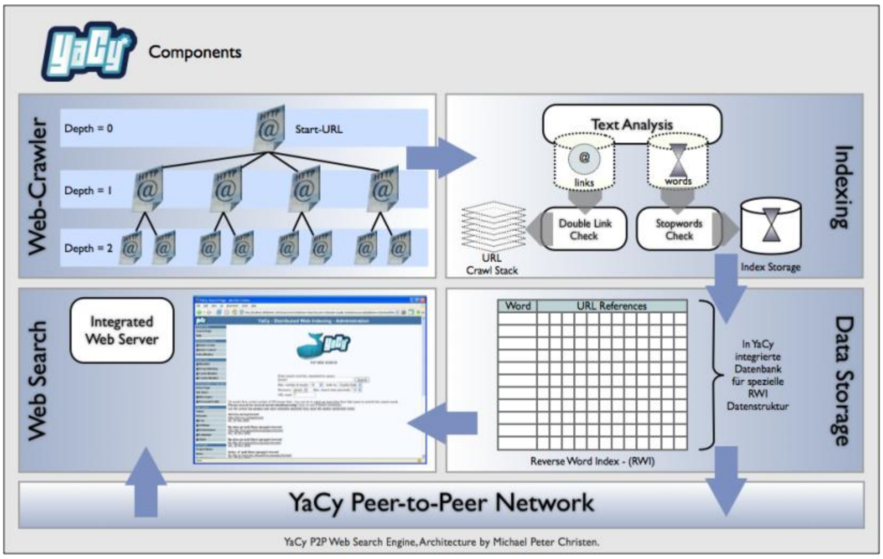


Fig. 3. Distributed indexing process

The figure 3, to have the main components of YaCy, and the process that exists among the web search, web crawler, the indexing and data storage processes.

5 Conclusions and ongoing work

In this section present some the conclusions and results that are expected of project and the future work.

1. Index all content of Wikipedia.
2. Storage this content.
3. Present the content of Wikipedia by topic in a web site.
4. Use a tagged of text for share the information with tags.
5. Present the module and their code on a web site
6. Share *knowledgebase* extract of Wikipedia

References

1. Definition of knowledgebase http://searchcrm.techtarget.com/definition/knowledge-base
2. Alarcón, R., Sierra, G., Bach, C. (2007). “Developing a Definitional Knowledge Extraction System”. En Vetulani, Z. (ed.), Actas del 3er *Language & Technology Conference. Human Language Technologies as a Challenge for Computer Science and Linguistics*. Poznan, Universidad Adam Mickiewicza: pp. 374-378.
3. Google Hacks, Second Edition, 2004, O’Reilly Media.
4. S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu. OpenDHT: a Public DHT Service and its Uses. SIGCOMM' 05, Philadelphia, Pennsylvania, USA, august 21-26, (2005).
5. http://www.jxta.org (2010).
6. http://yacy.net/ (2010).
7. http://www.twine.com/ (2010).
8. Query Processing on the Semantic Web Heiner Stuckenschmidt, Vrije Universiteit Amsterdam
9. http://www.alexa.com/siteinfo/google.com+yahoo.com+altavista.com (2009)
10. http://searchenginewatch.com/showPage.html?page=3630718 (2008)
11. http://research.google.com/university/search/ (2010)