SHUTTLE: VERTICAL CRAWLER'S DESIGN AND IMPLEMENTATION

Julio Cabrera-Corraliza

Escola Politècnica Superior d'Enginyeria de Vilanova I la Geltrú

***Abstract.* Shuttle is a complete crawling system designed and implemented to carry out the whole process of extracting information from specific websites and bringing it to a third application. This software has unique features of its kind, such as auto-generated code crawlers, capable of providing different behaviors or providing the system a distributed vision using multiple devices. This system is the result of five months of work and research, starting in summer 2012 and ending in January 2013.**

Keywords: Bots, Crawler, Data Mining, search engine, vertical search engine, TCP

Introduction

A Brief History of Information

Information is the result of the data treatment process. Information treatment could be thought of as a new science, but it's inherent to humans.

For example, in prehistoric times, data about their environment was represented in the form of cave paintings.

Further along the time, since that age until now, humans have improved the technology used for data representation and treatment.

Throughout the years humans have been relentless in their efforts to improve the way information is stored. In Egypt, for example, there was one of the biggest and most famous libraries: The library of Alexandria's –which had one million papyrus [1]-.

Before, the biggest amount of information and culture was stored in monasteries. The monks were the guardians of the books and knowledge.

The invention of Guttenberg's press, in XV Century, made it easier to distribute information among people and the access started to be public. Later, Antonio Meucci and John Baird made some improvements in this scope. The Italian invented the phone (1876) [2], and the Scottish man was able to managed to broadcast the first television signal between London and Glasgow (1926) [3].

In the 1940's, computational science was born with the creation of the first computer.

Since that moment, the treatment, process and dissemination of information is still improving.

Now, we have our particular Library of Alexandria's. Internet can provide you billions of documents. We use the Internet to shop, watch movies, and above all, to search information.

Search engines are the main tool to search whatever we want from the Internet. Many of them use internal software to crawl the Internet. They search information to make their databases. Search engines with this software allow you to look up any kind of information; and all this is done it with one mouse click.

Crawlers and Search Engines

There are several kinds of search engines. The most famous are the general search engines like Google or Yahoo! They have crawling software to make their database system.

The main parts of a general search engine are:

- **Crawler:** Also called web spider. It crawls the Internet searching information to save it in a database. Normally, it saves the URL and some key words that define the web pages. GoogleBot, BingBot y Yahoo Slurp! are the most famous crawlers all around the world.

- **Indexer:** With this tool, the information is uploaded to his database or persistence system.

- **Query Processor:** It is the part of a search engine closest to the end user. It attends user's petitions to look up information in a database of a search engine.

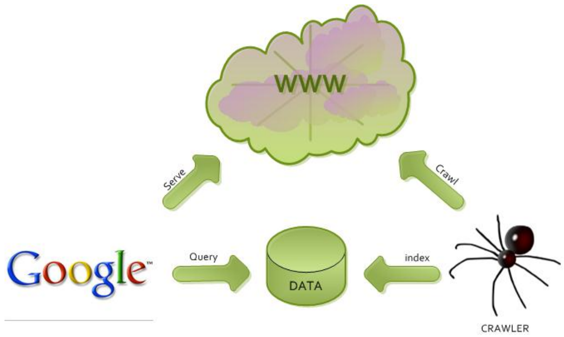


FIGURE 1. GOOGLE AND HIS CRAWLER

As aforementioned, there are other types of search engines. There are Directories, Verticals, and metasearch engines.

Directories do not have crawling software. In the harvest of information crawlers are not involved. The information is gathered by direct human intervention.

Metasearch engines do not collect the information by themselves. They show the results directly from other search engines.

The last kind of search engine is Vertical or Thematic search engine. There are two subclasses classified by the type of technology to crawl and collect the information:

- **Focalized Vertical Search Engine:** This type of vertical crawler is similar to a general search engine. They use the same technology, but they do not crawl all the World Wide Web, they assign a little group of specific web sites to crawl. These sites are directly selected by the administrator of the search engines. These pages are called seeds, because they are the first pages to crawl. One example is Dale ya! (downloads) or IMDB (movies).

- **Comparator Vertical Search Engine:** They are usually the central concept of a business, dedicated to compare products of the same thematic. This type of search engine can show more specific information than a general engine. They offer travels, hotels, insurances etc... Some examples are Trivago (Hotels), Last Minute (Travels) or Rastreator (Insurances).

Shuttle: Our Vertical Crawler

Shuttle is a complete system designed and implemented to solve the whole process of extracting information from specific sites and bringing them to a third application, such as a thematic search engine, or directly to users.

Why?

There are some problems to build a comparator crawler with the objective of be configured with a change of thematic extraction. The differences of the information format contained in different sites makes difficult to a vertical crawler been adapt to different topics.

The difference between each extraction of different thematic is big. It's not the same recollect dates for travels than compare price insurances.

How does Shuttle resolve it?

Shuttle provides to a search engine with a crawling system. It's composed by three applications that, further, resolve observed difficulties from the problem posed: we can wake changes in the theme of the crawler

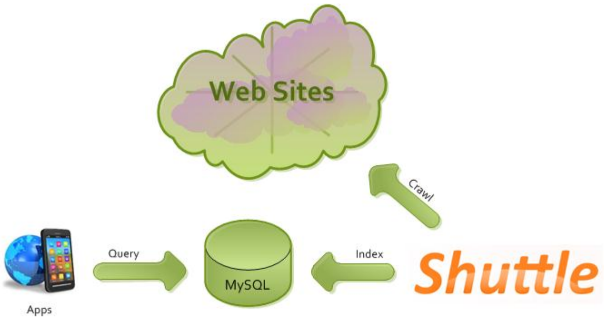


FIGURE 2. SHUTTLE EXTACTION PROCESS

The administrator of third party application can configure Shuttle to extract specific information from Web Sites and use it on his application.

Our system provides results in a Comma Separated Values file -one standard plain text persistence files- and, optionally, it can bring this data to a MySQL Data Base.

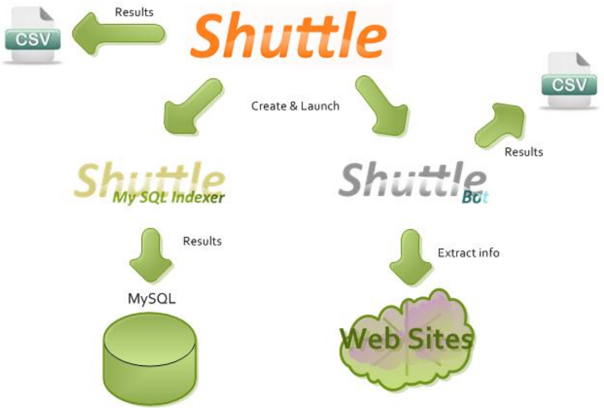
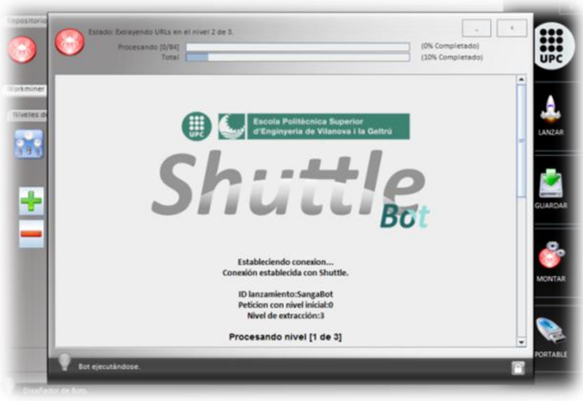


FIGURE 3. SHUTTLE ADVANCED EXTRACTION PROCESS

The three applications implicated in the process are:

- **ShuttleAdmin:** This application is coded in Java 7.1 language. It has the responsibility of managing the whole system. It has some features: launchings Bot management, a Bots Designer or a TCP server, to control the connections with the other applications.

- **ShuttleBot:** It's a configurable web application used to extract information from a specific site. This application is designed by a user with a Shuttle tool and is generated and executed by ShuttleAdmin.



Estado: Extrayendo URLs en el nivel 2 de 3.

Procesando [0/84] (0% Completado)

Total (10% Completado)

Estableciendo conexion...
Conexión establecida con Shuttle.

ID lanzamiento:SangaBot
Peticion con nivel inicial:0
Nivel de extracción:3

Procesando nivel [1 de 3]

Bot ejecutándose.

FIGURE 4. ONE SHUTTLEBOT WORKING ON SHUTTLE SYSTEM

- **Shuttle MySQL Indexer:** It's the third application of the system. It was conceived to bring the data to a final application, particularly implemented to allow an application to do queries from a search engine or similar system. This application is generated by ShuttleAdmin when you configure the database connection.

Design

The design of this system was done in three phases:

- **First phase:** It's the core of the project. Here we designed the minimum modules to make a functional vertical crawler. These modules are: Administrator and some implemented bots.

- **Second phas**e: Once we have found the way to provide different behaviors to bots, we add a new module to design the auto generated code bots. These bots are called ShuttleBots.

- **Third phase:** Another extension was made to provide other application for indexing the information to a MySQL Data Base. Finally, we designed and implemented one TCP server to allow for an http-independent communication between the Administrator and ShuttleBots.

Frontiers Algorithm

This algorithm is the responsible of ShuttleBot's web navigation. It builds urls indexes called frontiers. It has filters to reduce the number of pages to visit and the total time of processing.

Workminer Algorithm

The Workminer algorithm complements the last explained algorithm. This algorithm extracts the information of each page of one frontier of links. It recognizes patterns using regular expressions (REGEX) [7]. This algorithm is configurable for each pattern, the user can choose between 4 ways:

- **Default:** Extract data in the body page.
- **URL:** Extract data in the URL.
- **Mirror:** The user can choose to use the result of a previous pattern to complete a new pattern.
- **URL Replace:** User can change part of the URLs to extract.

Bot Server and portable Apps

The Bot Server is an administrator's module to manage an independent http connection to control the scripts generated by Shuttle. This module can support multiple bot connections at the same time using TCP protocol.

The user can distribute the work of our system. The scripts generated by Shuttle can be located in any computer with an open HTTP and TCP connection.

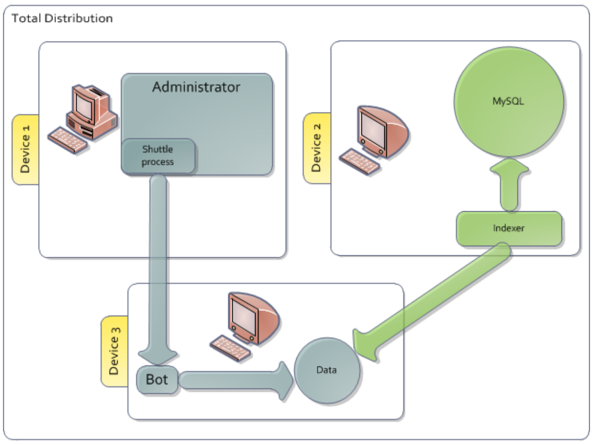


FIGURE 5. TOTAL DISTRIBUTION OF SHUTTLE APPS

Implementation

This project it's implemented in Java 7.1 and PHP. The first language allows us to work with an objected-oriented programming and multitier architecture, essential to our design. PHP allows us to replicate the code and provides good connection libraries.

ShuttleAdmin is composed 6708 lines of code distributed in twenty-seven Java classes. ShuttleBot and Indexer are codified with 679 lines of PHP code.

Multitier Architecture

The Java classes are classified in two layers:

- **Business layer:** it's responsible for the logical process of Shuttle. It's divided in two tiers: Services tier and Persistence tier.

- **Presentation layer**: Here are the classes that build the interface. They are the medium used to communicate the user with the main controller of the business layer. This controller starts the usage cases of the system.

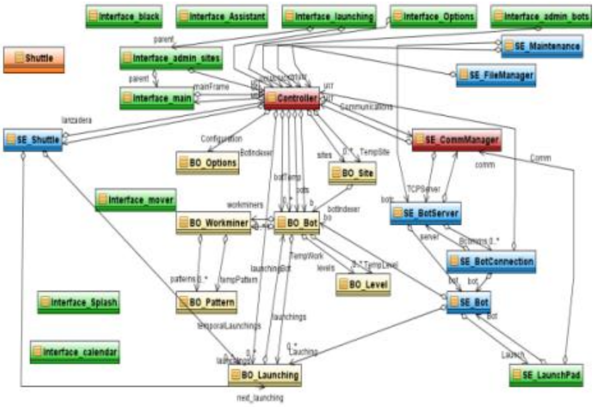
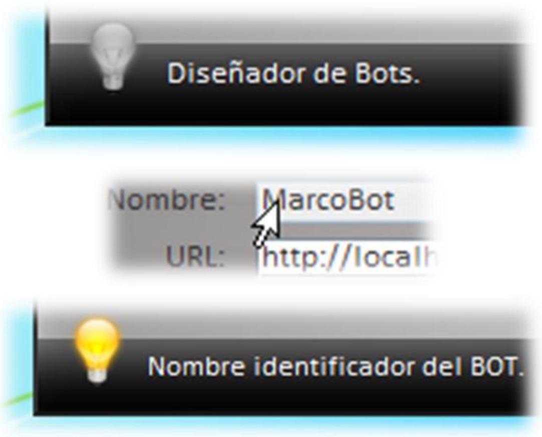


FIGURE 6. MINIATURE OF JAVA CLASS DIAGRAM
(IN GREEN, THERE IS THE PRESENTATION LAYER; THE REMAINING IS THE BUSINESS LAYER)

Interface

The interface of Shuttle is implemented with the aim to improve the user experience. This type of software is used by technical people, so we provide wizards, intuitive windows, and a help panel that describes each button, panel or field of Shuttle's interface.


Diseñador de Bots.

Nombre: MarcoBot

URL: http://localh

Nombre identificador del BOT.

FIGURE 7. HELP PANEL

Test

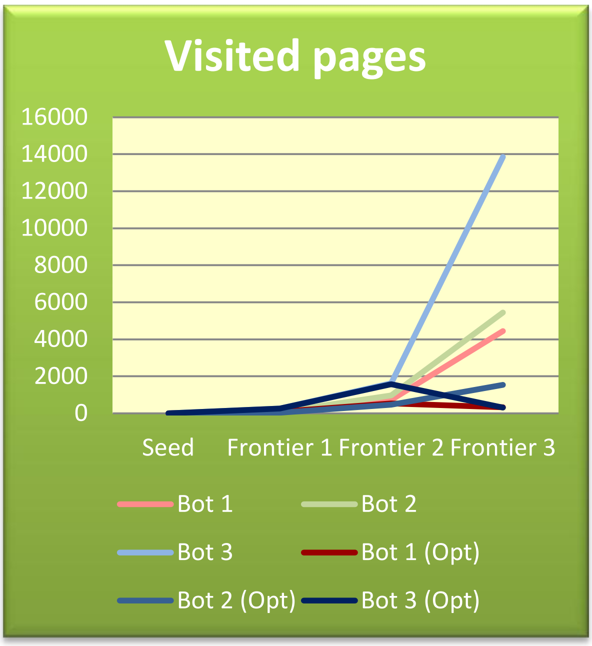
We have made various tests: We captured a TCP packet, checked the timing of the launch of bots and built an experimental vertical search engine to check the integrity of the data and proposed algorithms.

Buscar zona: kilimanjaro

FIGURE 8. EXPERIMENTAL SEARCH ENGINE

In the chart below we reflect one of the results of the experimental testing "Trekking & Adventure" search engine. In this case we have optimized three ShuttleBots (dark) and compared them with other three ShuttleBots without setting their browsing behavior (clear). The difference between the pages visited, and therefore the processing time, is considerable.

With our Frontiers Algorithm well configured we experience better results in terms of time and resources:



Conclusion

The main goal of this project was to design and implement an application system capable of extracting information from websites. This application system has been named Shuttle, and is part of the family of software classified as crawlers. The crawlers provide information, typically web applications as generalist or thematic search engines, such as Yahoo! or Trivago, respectively.

The project was implemented in five months, starting in summer 2012 and ending in January 2013.

Shuttle is the result of work and research evolved over half a year with the aim to get a thematic crawler. During these five months, the original project has expanded with new features in order to propose a thorough solution such as possible.

Reference & bibliography

1. "Archaeologists have found what they believe to be the site of the Library of Alexandria", BBC NEWS, *http://news.bbc.co.uk/ 2/hi/science/nature/3707641.stm*, May 12, 2004.

2. “Garibaldi-Meucci Museum”, GARIBALDI-MEUCCI MUSEUM, *http://pub1. andyswebtools.com/cgi-bin/p/awtp -home.cgi? d=garibaldi-meucci-museum*, July 30, 2002.

3. “John Logie Baird (1888 - 1946)”, BBC – HISTORY, http://www.bbc.co.uk/ *history/his toric_figures/baird_logie.shtml*, January 2013.

4. “How To Build A Basic Web Crawler To Pull Information From A Website”, James Bruce, *http://www.makeuseof.com/tag/build-basic-web-crawler-pull-information-website/*, December 10, 2010.

5. “Crawling de Hidden Web”, Ricardo Baeza Yates, *http://www.ciw.cl/wordpress/wp-content/uploads/2008/08/capitulo2.pdf*, 2006.

6. How To Build A Basic Web Crawler To Pull Information From A Website”, James Bruce, *http://www.makeuseof.com/tag/build-basic-web-crawler-pull-information-website/*, 10 de Decebmer 2010.

7. “Mastering Regular Expressions”, Jeffrey Friedl, O’REILLY MEDIA, January 1997, 368 pages, ISBN 1-56592-257-3.

8. “Computer Networking: A Top-Down Approach”, James F. Kurose y Keith w. Ross, 5ª Edición, PEARSON ADDISON-WESLEY, 719 pages, ISBN 978-84-7829-119-9. March 31, 2009.