WEB CRAWLER BASED APPLICATION

Autor: Mihai IACHIMOVSCHI
Coordonator: Mihail KULEV

Universitatea Tehnică a Moldovei

Abstract: *This article provides some basic key points about simple and complex web crawlers, xml parsers and databases. Also there are some general overview of the architecture, implementation, optimization and usage of the web crawlers. The purpose of application is to get unorganized open data from different web sites on daily basis, to filter and organize it, and to store it in a centralized database on a server for future usage. The data processed by application is currency rates from all commercial banks of Moldova. The domain of usage very large, from single price calculation to long term review of the trends of currencies.*

Key words: *currency, database, parser, trends, web crawler.*

5. Intro

Repetitive tasks are boring and doing them manually may include some severe human errors, that's why, people from Computer Science and Engineering in general, tend to automatize things. Automation is great because it frees up your mind to do what it does best, it lets you to focus on really important tasks.

6. What is a web crawler?

A web crawler is a program that browses the web in a methodical manner in order to get raw data. It collects data from different sources in a single place. The unstructured data is transformed (parsed) into structured data that is, eventually, persisted in a database. The term "unstructured" can be a little ambiguous, because, the data is ordered in a visual way, in a pattern from which we can extract useful data, so, let's call it, semi-structured raw data.[1]

Web crawlers, in a common sense are used vastly in search engine indexers and aggregators. This process is also called spidering, because is a recursively one, i.e. each browsed page provides new links that are queued for future crawling.

In general case, a recursive crawler works using this basic algorithm:

```
Init:
    linksList = starting element
    doneURLs = empty list
Loop:
    url = linksList.pop()
    page = Download(url)
    doneURLs.push(url)
        for each URL in page do
        if not doneURLs.has(URL) do
                linksList.push(URL)
```

In this case, the selection policy is focused only on few URLs, because is known clearly what pages to copy, and is known precisely what regions of the page contains useful data for us. Using this rules gives us the possibility to extract easily data and persist it into the database.

Also, the application is not a violent crawler that downloads thousands of pages per day. It makes just a few requests each morning between 8:00 AM – 9:30 AM, to each commercial bank of Moldova, downloads the content of this pages, and transfers them to a ETL-like routine that extracts chunks with data, transforms them in order to fit the technical requirements and loads the obtained data into the database.

7. What is ETL (Extract – Transform – Load)?

ETL is a concept in the database usage that involves three basic steps:

- Extracting data from outside sources;
- Transforming it to fit operational need;
- Loading it into the end target, i.e. our database. [2]

8. General features of the application

The application collects currency exchange rates on daily basis. The goal of the program is run daily, in order to have the freshest data. Every morning, when, by the rules of National Bank of Moldova, all other commercial banks are forced to publish their rates on their official web sites, the application surfs through all these sites and saves the generated html code.

From the saved code, other component of the program grabs raw data. A XML parser accomplishes this. For this task can be used Regular Expressions in order to get data from specified patterns, but, a better approach is to use parsers, because they are faster, easier to maintain, and more efficient for big amounts of data.

After that, there is a sanitizer routine that checks if the obtained data have the form of the expected one. If we have good data, it is inserted in the database, otherwise, there is generated an email alert, to check what happened.

And, after a while, if the program works normally, a fully populated database is got from a process that works without any human involvement and maintains the database up to date.

9. Why do we need an application like this?

Having all raw data from a long period of time, we can:

- Analyze long term trends of the currency;
- Compute a average difference of currencies trends;
- Plot graphs;
- “Predict” near future trends;
- Use the values from a centralized database in any other purpose;

10. Outro

In Figure 1 and Figure 2 is represented the trend of Euro and USD for the period of February 2012 – November 2012. Obviously there is no possibility to do representations like this without having a fully functional up to date database. The deduction is simple, in a data oriented project, focus on automatizing data population rather than investing in some empiric or manual ways of getting data.

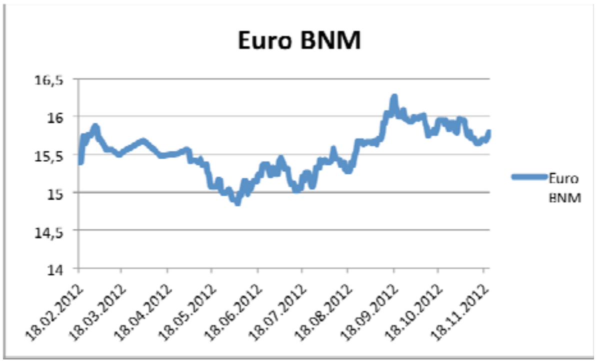

Figure 1

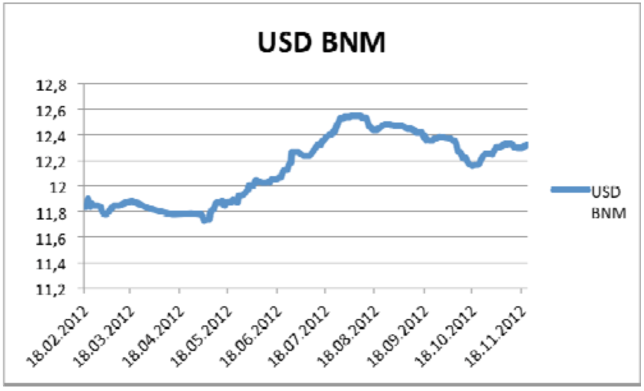

Figure 2

References

1. http://en.wikipedia.org/wiki/Web_crawler
2. http://en.wikipedia.org/wiki/Extract,_transform,_load