Web scrapping from **400,000 Product IDs** provided by the client

Web scrapping from 400,000 Product IDs provided by the client

Requirement

The client has given us more than 400,000 Product variables from which we had to scrap all the required data that has to be uploaded to a database. The data that we acquire has to be compared with various factors so that a vivid analytical view can be possible.

Our Solution

We built a robust system that can effectively crawl through all the link provided by the client or even a website and can scrape data from it as fast as possible.
The structure that we have developed is scalable and cost-effective compared to the other solutions available in the market.

Features of our solution include,

- Crawl ecommerce websites with impeccable efficiency
- Fetch data incorporated in product pages
- Fetch images
- Errors occurred are logged so that they can be reviewed later on
- Data will be added to the DB
- Fresh and clean data was delivered to the client on timely basis
- Constant data delivery was ensured
- Cost-effective

Common issues faced in web scrapping

First of all, we had two choices in front of us either to build a scrapper for each website or to make a complex structure that can be put to use for many websites.

- First option is indeed hard as maintaining scrappers, many of them is an unimaginable task
- Making a complex structure is no easy task. Accuracy has to be more than 90% for the system to be effective. For that to happen incorporation of multiple technologies was needed
- Clients must be able to view the quality and progress of our work in real-time.
- We had only 4 proxy server to work from

- If there is any kind of error in the link provided, the system has to automatically skip it and retry again after some time
- The time frame was limited

What we did

- A **DB** table was given to us by the clients which have either SKU's, Product ID or EAN (European Article Number)
- For repetitive tasks, a program was written in PHP cron so that links will be attached to the specific ID's given by the client.
- The URL will then be passed through PHP CURL and a proxy server so that HTML can be fetched continuously without any disruption
- Fetched HTML code is passed through PHP Simple HTML DOM to acquire the required data
- Acquired data has been added to a spreadsheet in DB with values 0 and 1 (o indicates the task not completed and 1 indicates vice versa)
- Another program was written in Cron to make the scrapping process faster. Forking was done to spawn sub processes from a parent process so that both programs can run simultaneously.
- If there is any dead link or there is an error in it, the program will skip to another link after a specific amount of time. Meanwhile, the subprocess will continue and will again recheck the link to ensure maximum efficiency
- An interface was created in .NET for us to review the progress of the scrapping as well as an analytical graph for the comparison of the products to ensure that there is no flaw.

Application

- Real-time monitoring of competitors price
- The main tool for price comparison websites
- Scattered data can be manifested and brought to one place

- Performance of a particular product can be analyzed
- You can have a look at the price of supplier websites so that profit margin can be set
- Public review of a product can be analyzed
- Price of products can be updated by the latest trend

Feel free to contact us

➔ ***Pune***

➔ ***Kalas road, Vishrantwadi, Pune, Maharashtra-411015,***

➔ ***+91 81 1386 1000***

➔ ***info@probytes.net***