ISSN: 0974-5823 Vol. 7 No. 6 June, 2022

International Journal of Mechanical Engineering

WEB DATA SCRAPPING IN BIG DATA APPLICATIONS USING EVENT-DRIVEN ARCHITECTURE

Midhu Bala G[1]

Research Scholar

Department of Computer Science

Madurai Kamaraj University, Madurai

Dr.K.Chitra[2]

Assistant Professor

Department of Computer Science

Govt., Arts College, Melur, Madurai

Abstract

A massive data in Web, twitter and social media plays a vital role in business to acquire substantial value that primes to take decision & improve their sales. As an outcome, Extraction of data through web becomes an essential tool for collecting and turning semi-structured data into valued information. Nevertheless, the most important challenge in internet is growing rapidly and data grows more and more in social media, scraping the web pages becomes difficult in the dynamic environment. This study proposes, builds a big data design with WDSEDA – (Web Data Scraping Event-Driven Architecture) with Node.Js agent and asynchronous I/O calls with added URL parameter to deploy and scrap data, information automatically from various web sources. The proposed design also extracts real time data using asynchronous I/O calls concurrently from external system by following data mining guidelines and data transformation policies. The data extraction engine automatically crawls, scrapes and converts un structured, semi-structure data into structured relational data. Our proposed design has effectively extract data from online web pages, sources and documents using node.Js event-driven applications.

Keywords*— Information Data retrieval, Web Data Scraping, JavaScript Web application; JSON, Asynchronous I/O calls;*

I. INTRODUCTION

In digital era the development of data such as text, photos, videos, feedback, reviews become tremendous in WWW, twitter and social media. This circumstances create an opportunity for analyzers, users to get use of available information to use in different ways, specially data science, analytics and decision. Through analytics of big data from internet, twitter and social media, online trade, e-commerce and e-business increase their sales, customer relationship and achieve high profit. Though, the information on the internet is not in a structured form for processing analytics and decision making. Still, numerous data sources are openly available as Web services, several data sources are still not accessible through a programming interface [1]. As an outcome, the tools used for data extraction from web sources, have become an significant tool for collecting data [1].

An extraction of web data system is a software application that extracts data from web pages automatically, provides the extracted data to a database or some other application. This data extraction process involves seeking, fetching the data and extracting it. Fetching is a process of downloading a web page and extraction of data is downloading data after fetching. The downloaded documents are then formatted, transformed into structured data that is user required format. The process of fetching data and extracting it is usually done automatically and repeatedly in order to deliver extracted data into a spreadsheet, database or some other application [2].

The main challenges faced in Web Data Extraction is dynamic and real-time. Key feature that affects the structure of Web documents is evolving of web development technologies. Developing of JavaScript frameworks in Web development has expressively changed the technique of data embedding and rendering process of Web pages.

The data extraction process from world wide web first starts with sending a request to the target server, then the server response and renders data in the form of HTML document to the client. This is the technique followed to extract data from web pages by Web Extraction tool. The Node.js tool harvest data from Webservers through asynchronous JavaScript I/O calls.

In this proposed study, we design a Developmental workstation, Web Data Extraction system to extracts data from JavaScript Web applications. This proposed system is tested by extracting product information from an online shopping Website.

II. BACKGROUND AND RELATED WORK

A. Web Data Extraction

Extraction of Web Data has been used in variety applications such as data analytics, document analysis, business intelligence,social media, data science and so on. Web data extraction system is developed for mining unstructured data from Social Media and Online Social networks, emails, business forums, web documents blogs [3]. In this paper, we focus on extracting data from Web pages that are enormous of Unstructured information in HTML pages. To competently pull outdata from HTML documents, the existing system take on the broad class of techniques that is processing text, DOM parsing,natural language processing and machine learning [4, 5].

The proposed method of Web Data Extraction consists of two parts applied with different techniques. The first phase is algorithmsfor extracting data from HTML documents. Since a Web document is a hierarchy of HTML elements that are usually represented as DOM (Document Object Model), most of data extraction systems rely on tree-based techniques i.e. addressing,matching and weighing of tree nodes [6]. The second part is called Wrapper, which is a procedure that implements one or multiple data extraction algorithms [7].

B. Big Data technology

Our choice of technology sets a foundation for future research, and helps for further analysis. Extracting and Processing I/O of hundreds of pages of web products is manageable on standard desktop with conventional network connections. Though, managing huge number of web page processing, including cleaning data during runtime does require ascendable technology. Big Data technology has become an essential for research in future, so we lay a base here for extracting data for the research.

III. WDSEDA - Proposed Architecture

A series of asynchronous I/O algorithms with node.js were custom-developed for acquiring web data and pre-processing Amazon.com data. The system sits within a Web and Social Media Big Data client–server architecture, integrating various open-source server technologies used by large corporations (e.g. LinkedIn, Yahoo, Microsoft, eBay, etc.). The system consists of 6x Linux Ubuntu 64bit Virtual Machines (VM) on 2× HP DL388p physical servers. The physical system is horizontally scalable as needs arises for additional VMs. Due to the asynchronous I/O nature of our algorithms, as data comes in, they are efficiently stored within MongoDB, a cross-platform NoSQL database that is horizontally scalable prior to synchronous extraction later into Comma Separated Value (CSV)file for our Eagle Search analysis. Our asynchronous algorithms are coded in server-side JavaScript via Node.js. Node.js is an event-driven, non-blocking I/O model built on Google's V8 JavaScript engine. Due to the asynchronous nature of Node.js, it is capable of managing real-time, data-intensive applications such as what we have in the present research. Our algorithms are developed and deployed on a Dell T3600 Tower Workstation with 64 GB of RAM, 6 cores 12 threads, with Quadro K4000 and Tesla K40c GPGPU. The Tesla K40c was prepared for parallel processing needs, however, it was not utilized as the data were not sufficiently large to require multicore processing.

We first identified product links before deploying our Web crawlers, once our crawlers have reached a sufficient population of product links, we deployed our asynchronous Web scraper agents, which parses the incoming HTML data and extract targeted elements via regular expressions in real-time. Incoming data are immediately stored within the MongoDB server, and a CSV file was generated after the scraper agents completed their jobs.

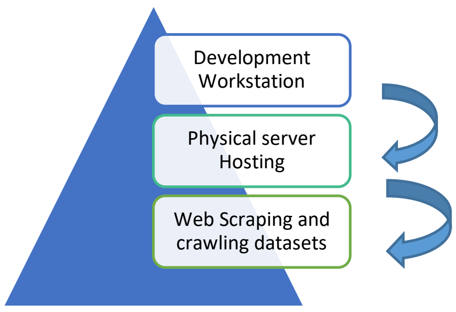


Fig.1 WDSEDA - Architecture

The process has three steps:

(1) Developmental workstation where our Node.JS agents are deployed for scraping the web using asynchronous I/O calls.

(2) Physical server hosting Ubuntu 64bit virtual machines, and where data are stored and horizontally scaled.

(3) Completed Web crawling and scraping data-sets are converted into comma separated values for eagle search analysis.

1. Developmental Workstation

Asynchronous I/O Algorithms

- Interact with all external System
- Parallel function handles several request concurrently and receive the response concurrently
- Waiting time can be overlaid
- custom-developed for acquiring and processing Amazon.com data.
- Asynchronous algorithms are coded in server-side JavaScript via Node.js.
- Node.js is an event-driven, non-blocking I/O model.
- Due to the asynchronous nature of Node.js, it is capable of handling real-time, data-intensive applications

2. Physical Server Hosting

- The physical system is horizontally scalable as needs arises for additional VMs.
- Due to the asynchronous I/O nature of algorithms, as data comes in, they are efficiently stored within MongoDB
- MongoDB - a cross-platform NoSQL database that is horizontally scalable prior to synchronous extraction.

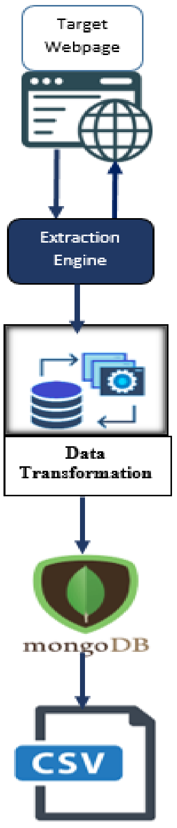
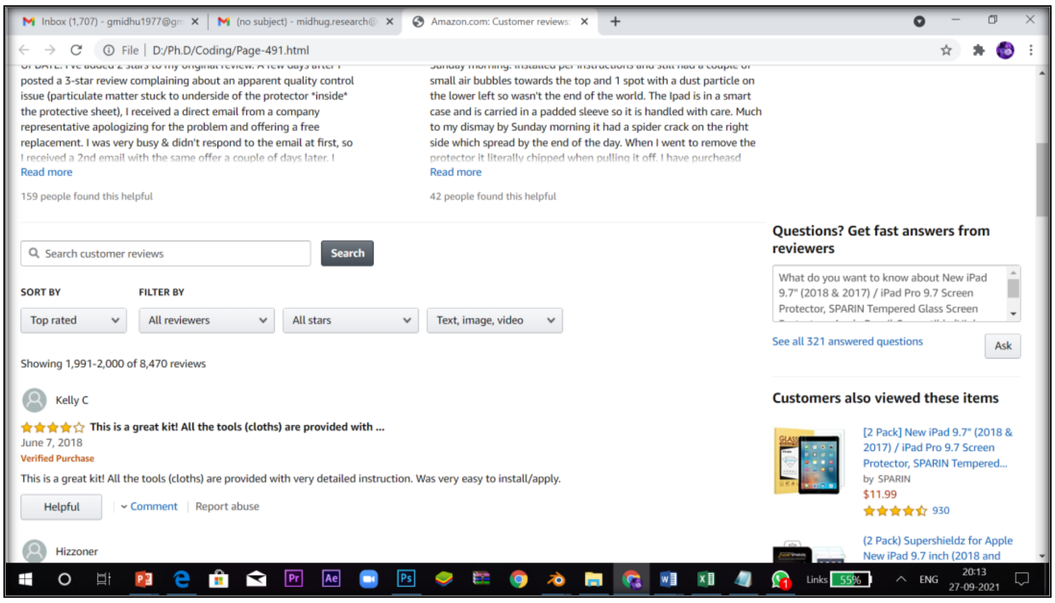


Fig.2 WDSEDA - Proposed Architecture Overview

IV. IMPLEMENTATION

A. System Development

To enable users to easily extract data from various Web documents, we tested our solutions with the real-world data the proposed system is developed using Node.js asynchronous I/O calls. Fig.3 shows Extracted page from amazon website, MongoDB is used to store huge amount of real time data.



posted a 3-star review complaining about an apparent quality control issue (particulate matter stuck to underside of the protector *inside* the protective sheet), I received a direct email from a company representative apologizing for the problem and offering a free replacement. I was very busy & didn't respond to the email at first, so I received a 2nd email with the same offer a couple of days later. I

Read more

159 people found this helpful

small air bubbles towards the top and 1 spot with a dust particle on the lower left so wasn't the end of the world. The Ipad is in a smart case and is carried in a padded sleeve so it is handled with care. Much to my dismay by Sunday morning it had a spider crack on the right side which spread by the end of the day. When I went to remove the protector it literally chipped when pulling it off. I have purcheasd

Read more

42 people found this helpful

Search customer reviews

Showing 1,991-2,000 of 8,470 reviews

Kelly C

★★★★☆ This is a great kit! All the tools (cloths) are provided with ...

June 7, 2018

Verified Purchase

This is a great kit! All the tools (cloths) are provided with very detailed instruction. Was very easy to install/apply.

Hizzoner

Questions? Get fast answers from reviewers

What do you want to know about New iPad 9.7" (2018 & 2017) / iPad Pro 9.7 Screen Protector, SPARIN Tempered Glass Screen

See all 321 answered questions

Customers also viewed these items

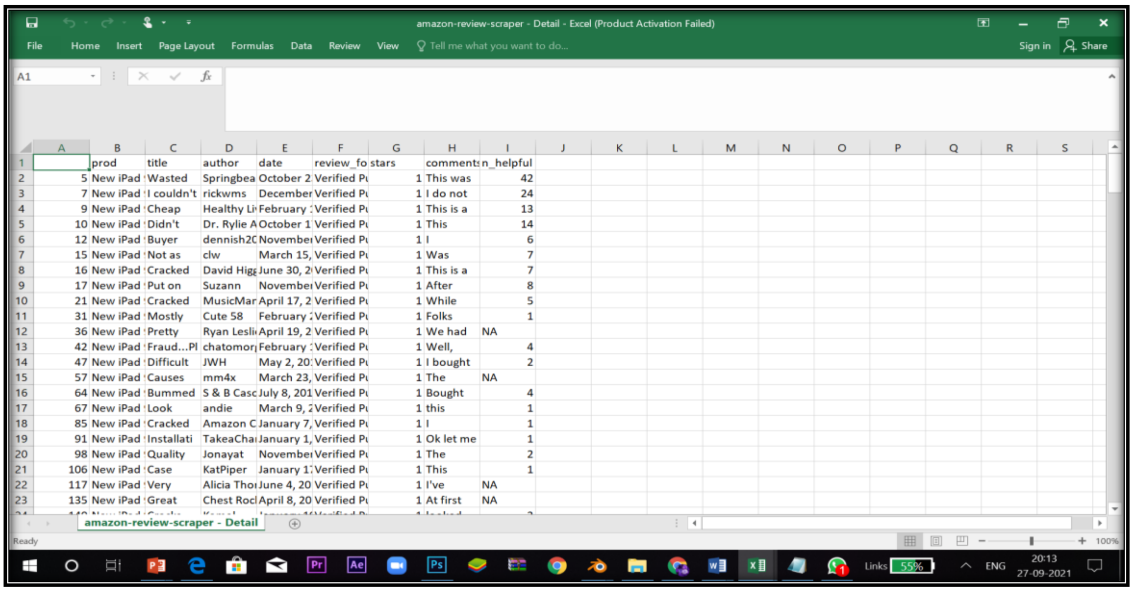
[2 Pack] New iPad 9.7" (2018 & 2017) / iPad Pro 9.7 Screen Protector, SPARIN Tempered...
by SPARIN
$11.99
★★★★☆ 930

(2 Pack) Supershieldz for Apple New iPad 9.7 inch (2018 and

Fig.3 Extraction of Data from Amazon Website

B. Experimental

Data Extraction: Product information is e*xtracted from amazon website:* This scenario simulates a data gathering task thataims to collect product id, information, reviewer, rating, comment from amazon Website. we isolate the data like items, selected items, default filter, implied items, arguments, data filters. The data is separated based on time such as month, day, quarter, year, weak. The final result of the proposed system is shown in Fig.4, the data filtered and stored as csv format.


amazon-review-scraper - Detail - Excel (Product Activation Failed)
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Tell me what you want to do...
Sign in
Share
A1
amazon-review-scraper - Detail
Ready

	A	B	C	D	E	F	G	H	I
1		prod	title	author	date	review_fo	stars	comments	n_helpful
2	5	New iPad	Wasted	Springbea	October 2	Verified Pu	1	This was	42
3	7	New iPad	I couldn't	rickwms	December	Verified Pu	1	I do not	24
4	9	New iPad	Cheap	Healthy Li	February	Verified Pu	1	This is a	13
5	10	New iPad	Didn't	Dr. Rylie A	October 1	Verified Pu	1	This	14
6	12	New iPad	Buyer	dennish20	November	Verified Pu	1	I	6
7	15	New iPad	Not as	clw	March 15,	Verified Pu	1	Was	7
8	16	New iPad	Cracked	David Higg	June 30, 2	Verified Pu	1	This is a	7
9	17	New iPad	Put on	Suzann	November	Verified Pu	1	After	8
10	21	New iPad	Cracked	MusicMar	April 17, 2	Verified Pu	1	While	5
11	31	New iPad	Mostly	Cute 58	February	Verified Pu	1	Folks	1
12	36	New iPad	Pretty	Ryan Lesli	April 19, 2	Verified Pu	1	We had	NA
13	42	New iPad	Fraud...Pl	chatomor	February	Verified Pu	1	Well,	4
14	47	New iPad	Difficult	JWH	May 2, 20	Verified Pu	1	I bought	2
15	57	New iPad	Causes	mm4x	March 23,	Verified Pu	1	The	NA
16	64	New iPad	Bummed	S & B Casc	July 8, 201	Verified Pu	1	Bought	4
17	67	New iPad	Look	andie	March 9, 2	Verified Pu	1	this	1
18	85	New iPad	Cracked	Amazon C	January 7,	Verified Pu	1	I	1
19	91	New iPad	Installati	TakeaChai	January 1,	Verified Pu	1	Ok let me	1
20	98	New iPad	Quality	Jonayat	November	Verified Pu	1	The	2
21	106	New iPad	Case	KatPiper	January 1	Verified Pu	1	This	1
22	117	New iPad	Very	Alicia Thoi	June 4, 20	Verified Pu	1	I've	NA
23	135	New iPad	Great	Chest Rocl	April 8, 20	Verified Pu	1	At first	NA

Fig.4 Converted into CSV format

V. Result and Discussion

The experiment was designed to extracts various product information from different URL by our system. We acquired result with huge number of records that is extracted and stored in Mongodb, and then it is converted as a csv file for further analysis. The evaluation result is shown in Fig.1 Our proposed system has successfully extracted the target data from the Web pages.

JavaScript page navigation: Normally, JavaScript Web applications uses lazy loading technique to navigate the web pages. The first set of data is retrieved by sending request and the remaining data are fetched by sending next request. We have added more URL parameter to fetch the data, to automatically navigate to the configuration to simulate a request that fetches least recent data. In order to automatically navigate the feature is further added in Extraction Engine.

1) ***Duplicate data:*** *The* frequency of extraction in the target Web pages cannot set perfectly this leads to creation of *Duplicate because of the same documents were repeatedly examined. So*, allow duplicate setting is added.

2) ***Limitation:*** Our proposed system is designed to extract the data from the various target Web documents. The target Websites exposed JSON data in their DOM which has to be ranked by the search engines, normal JavaScript applications extract their pages from server-side as ordinary client-server Web applications.

VI. CONCLUSION

This paper designed to extract data from various web sources through Node.js with asynchronous I/O calls. The proposed system scraps data from existing Web sites by following extraction rules, schedules, storage and data transformation. The developed system, Web data extraction engine fetches dynamic real time data through the headless browser with Node.js asynchronous I/O algorithm along with added URL parameter. The implementation was successful and yielded with the data from amazon online shopping website. This dataset can be preprocessed and analyzed with classification algorithms and optimization techniques for predicting product based on customer feedback, ratings & reviews.

REFERENCES

[1] "A Survey of Web Information Extraction Systems", C. Chang, M. Kayed, M. R. Girgis, K. F. Shaalan, C. Chang, M. Kayed, M. R. Girgis, and K. F. Shaalan IEEE Trans. Knowl. Data Eng., vol. 18, no. 10, pp. 1411–1428, 2006.

[2] "Web Data Extraction, Applications and Techniques: A Survey", E. Ferrara, G. Fiumara, and R. Baumgartner, ACM Trans. Comput. Log., vol. 1, June, pp. 1–20, 2010.

[3] "A Survey of Web Information Extraction Tools", N. Negm, P. Elkafrawy, and A. B. Salem, J. Comput. Appl., vol. 43, no. 7, pp. 975–8887, 2012.

[4] "Deep neural networks for web page information extraction," in IFIP Advances in Information and Communication Technology, 2016, T. Gogar, O. Hubacek, and J. Sedivy, vol. 475, pp. 154–163.

[5] "Extraction of Web News from Web Pages Using a Ternary Tree Approach," in Proceedings - 2015 2nd IEEE International Conference on Advances in Computing and Communication Engineering, ICACCE 2015, D. Laishram and M. Sebastian, 2015, pp. 628–633.

[6] "Using the DOM Tree for Content Extraction", S. López, J. Silva, and D. Insa, Electron. Proc. Theor. Comput. Sci., vol. 98, pp. 46–59, 2012.

[7] "Web wrapper induction: a brief survey", S. Flesca, G. Manco, E. Masciari, E. Rende, and A. Tagarelli, AI Commun., vol. 17, no. 2, pp. 57– 61, 2004.

[8] "Single Page Application using AngularJS", M. A. Jadhav, B. R. Sawant, and A. Deshmukh, in International Journal of Computer Science and Information Technologies, 2015, vol. 6, no. 3, pp. 2876–2879.

[9] "Full web stack No browser required", PhantomJS" [Online]. Available: http://phantomjs.org/.

[10] "Analysis on Incorporating the Influenced Factors for Online Product Sale by Review System and Promotional Strategy using Optimization of Online Promotion System", Design Engineering, vol 2021 Issue 7, pp. 14117-14132.

[11] "Comparative and Performance analysis of FS- Feature Selection Optimization Techniques for Promotional Marketing Strategy to Predict and Forecast Product Demand (PMSPF) on Dissimilar Datasets Using Deep Learning Techniques", Vol.7(Special Issue 2.Jan.-Feb.2022)