A Scalable End-to-end Web Application to Fix Broken Hyperlinks

Maximiliano Cárdenas, Eng. (c) [1], Roger Calderón, M.Sc. [1], and Javier Enciso, M.Sc. [1]
[1]Universidad de los Llanos, Colombia, {maximiliano.cardenas, rcalderonmoreno, jenciso}@unillanos.edu.co

***Abstract*– Fixing broken hyperlinks is a regular task performed by web content curators. Although there are several technologies addressing this issue, there is no evidence of a tool that incorporate all steps required to fix a broken hyperlink. BYE BYE 404 is a web application that automates this process. It detects broken hyperlinks and suggests how to fix them based on advanced information retrieval techniques. This paper presents an overview of the related applications, involved technologies and testing results.**

Keywords--Broken hyperlink, BYE BYE 404, IaaS, Web maintenance

I. Introduction

A broken hyperlink happens when the referring resource is no longer available, for instance, after a webpage changes from one location to another. This is a common issue for web content curators. In fact, estimation shows that the half-life for external hyperlinks is two years [10], making fixing broken hyperlinks a regular task for those responsible to keep up to date the information available in websites.

Fixing a broken hyperlink involves thee steps: i) detection, ii) correction, and iii) prevention. Each step involves a considerable amount of time. This time increases proportional to the number of webpages.

BYE BYE 404 is a web application capable to automate the three steps involved in fixing a broken hyperlink. It is scalable and offers an end-to-end solution for the maintenance of hyperlinks across websites.

BYE BYE 404's interface enable users to create an account, add websites, fix broken hyperlinks, among other features.

The remainder of this paper proceeds as follows: section II reviews related work; section III exposes the solution implemented to address the problem. Finally, sections IV and V present the results and conclusions.

II. Related work

Table 1 shows an overview of some popular related applications as listed in [1]. Although there are similar tools available, they lack in offering an end-to-end solution to the hyperlink decay problem.

TABLE I
POPULAR RELATED APPLICATIONS

Application name	Description
W3C Link Checker	Tool that looks for issues in links, anchors and referenced objects in a webpage, CSS style sheet, or recursively on a whole website. It is part of the W3C's validators and Quality Web tools.
LinkChecker	Free, GPL licensed website validator. A desktop tool that checks links in web documents or full websites.
SEO Spider	Desktop program that checks websites' links, images, CSS, scripts and apps to evaluate onsite SEO. This tool has free and paid versions.
Broken Link Checker	*Wordpress* plug-in that checks websites periodically to fix broken links and images based on user preferences.
Pinger	Firefox add-on to check for broken links on a website.
Online Broken Link Checker	Online tool to check for broken links on a website. The paid version allows checking recursively on pages inside the website's main page.
Xenu's Link Sleuth	Tool that checks websites for broken links. Verification includes hyperlinks, images, frames, plug-ins, backgrounds, local image maps, style sheets, scripts and java applets. It displays a continuously updated list of URLs, sorted by different criteria; reports are available at any time.
Google Webmaster Tools	Tool designed to inform webmasters about how Google bot is interacting with their website. It gives detailed information on broken links, popular keywords and visits through Google.

III. SOLUTION

The solution uses the Infrastructure as a Service (*IaaS*) paradigm, which is a type of infrastructure that is provided and managed online, and allows for easier and faster provisioning of resources to adjust them to the requirements of the demand.

IaaS reduces the costs and complexity that come with the manual administration of physical servers and other physical data-center infrastructure [2], since an external party provides technical support.

This project uses *Amazon Web Services* (AWS) cloud computing potential. AWS is a service platform provided by *Amazon Inc.* (NASDAQ: AMZN) that offers cloud computing, database storage, content delivery and other features to help enterprises grow [3], the services used in this project are Elastic Beanstalk [4], S3 [5], RDS [6], EC2 [7], SES [8], and Route 53 [9]. Fig. 1 shows an overview of the proposed application architecture. The solution uses AWS Elastic Beanstalk as application container. Within, a Django

Digital Object Identifier: (to be inserted by LACCEI).
ISSN, ISBN: (to be inserted by LACCEI).

application implements the services to perform asynchronous administrative tasks and the algorithms to detect, correct, and prevent broken hyperlinks. In addition, there is an integration with external payment platform to receive contributions from paid users.

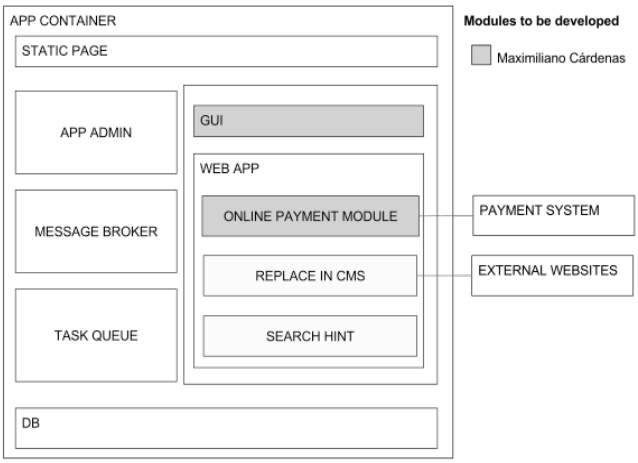

Fig. 1 Architecture of the solution. *Django*[1] is a high-level *Python* Web framework. *Celery*[2] is the distributed task queue, and *RabbitMQ*[3] is the message broker. *Git*[4] is the version control system.

Fig. 2 shows the use-cases implemented in the solution. As seen, Users are able to log in and manage websites, pages, and broken hyperlinks. In a background process, the system analyses the pages to look for broken hyperlinks. Once found, another module looks for candidate solution out of an internal database and queries to external search engines and internet archives.

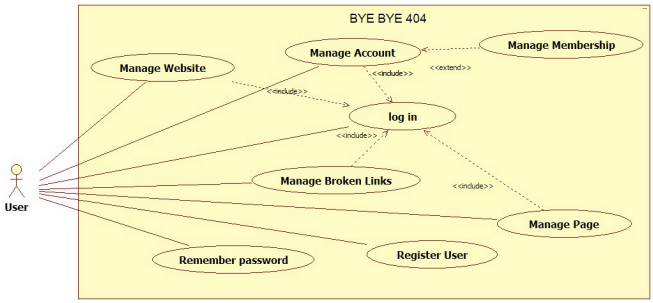

Fig. 2 Use-cases of the solution. Most of the use-cases relies on the log in use case to enable users to provision services within the system.

IV. RESULTS

A. *Interfaces*

Fig. 3 to 5 shows the interfaces for account created, account activated and Dashboard respectively. The front end uses HTML code minification and gzip encoding to reduce the loading time. JavaScript enables the interactivity of the interface components.

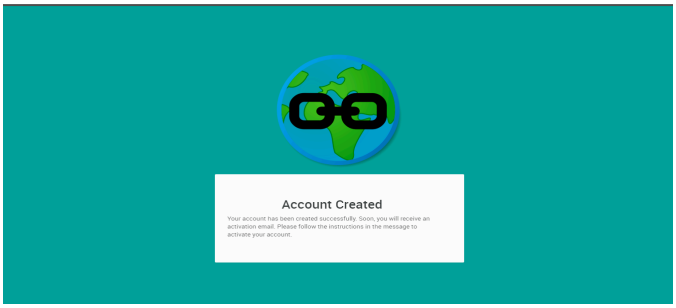

Fig. 3 Screenshot for successful account creation. HTML minification contributes to reduce the loading time for each page.

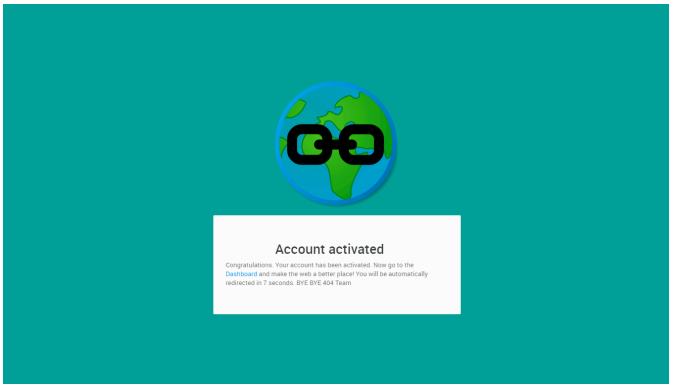

Fig. 4 Screenshot for account activation. The system notifies users by sending activation emails. Activation email expire every 24 hours to prevent automated account activation.

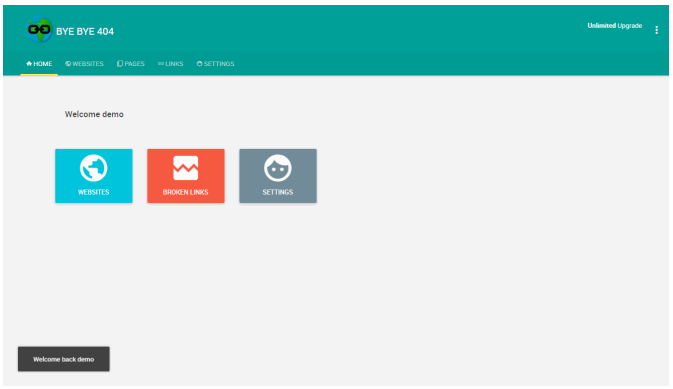

Fig. 5 Screenshot for Dashboard. From this point, users configure the analysis of websites. In turn, the system reports pages with broken hyperlinks and how to fix them.

[1] https://www.djangoproject.com/
[2] http://www.celeryproject.org/
[3] https://www.rabbitmq.com/
[4] https://git-scm.com/

B. Stress Test

Different stress test scenarios used *Apache JMeter*[9] to simulate the concurrent connection of *N* users, represented by *N* threads interacting with the system. By doing so, we ensure to replicate the exact conditions of the test and produce comparable results.

We present the results for *N*=100, *N*=50, and *N*=10 concurrent simulated users in Fig. 6 to 8 respectively. In the horizontal axes, we show six different methods to test. In the vertical axes, we plot the time in milliseconds the system takes to respond a request.

The test is developed for the following server connection points: to obtain the login form, to send the completed form to the server, to obtain the websites, to obtain the pages of the websites, to obtain the broken links and to visualize the account information.

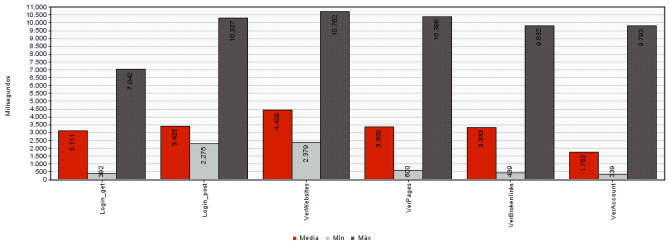

Fig. 6 JMeter test with 100 threads. On average, the system takes about 11 seconds to produce a response.

Fig. 6 shows the response time from the server. In this case, there are 100 concurrent threads and the system takes about 11 seconds to produce a response.

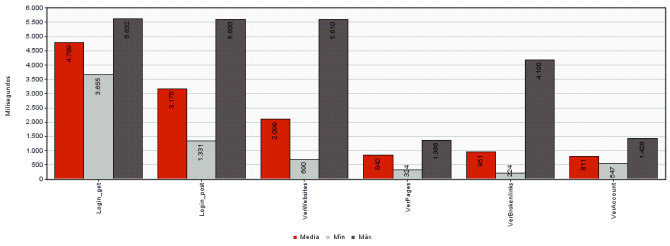

Fig. 7 JMeter test with 50 threads. On average, the system takes about 5 seconds to produce a response.

Fig. 7 shows the response time from the server. In this case, there are 50 concurrent threads and the system takes about 5 seconds to produce a response.

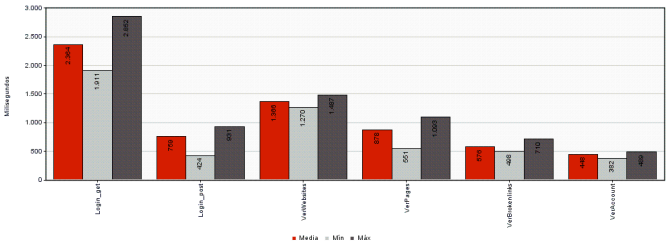

Fig. 8 JMeter test with 10 threads. On average, the system takes about 3 seconds to produce a response.

Lastly, Fig. 8 shows the response time from the server. In this case, there are 10 concurrent threads and the system takes about 3 seconds to produce a response.

C. Functional Test

We performed functional testing with *SeleniumHQ*[10] to check each one of the platforms interfaces and ensure they do not present any error. Fig. 9 show the integration of the testing tool with the browser.

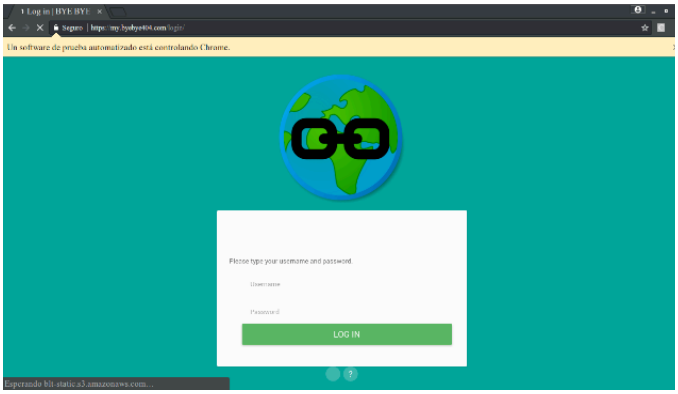

Fig. 9 Screenshot of the interface test using SeleniumHQ on Google Chrome. The testing frameworks integrates on the browser, making its usage a transparent process for test users.

These tests of functionality of the GUI prevent developers from having check the continuity of the work after making modifications in the user interface of the application. Testers record the test cases and executed with the new release.

D. Global Endorsement

BYE BYE 404 is a web application devoted to serve users around the world. Pilot customers include the websites of the European Southern Observatory (ESO), ESA/Hubble, and the International Astronomical Union (IAU). Fig. 10 shows a screenshot of the supporting technologies for IAU's website, highlighting BYE BYE 404.

[9] https://jmeter.apache.org/

[10] http://www.seleniumhq.org/

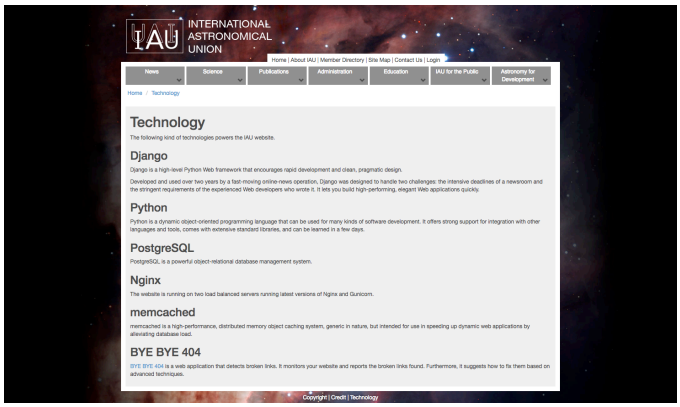

Fig. 10 Screenshot of the supporting technologies at IAU's website. Source: https://www.iau.org/technology/.

D. *Future directions*

New version of the application should include:

- Increase and organize information presented in the Dashboard.
- Implement alternative ways to check the integrity of a broken hyperlink to reduce the number of false-positives.
- Develop more plugins to support additional Content Management Systems (CMS's) and enable users to fix broken hyperlinks in one click.

V. CONCLUSIONS

- A scalable end-to-end solution to fix broken hyperlinks is a must-have in the Webmaster tool arsenal. It provides a consistent way to combat the hyperlink decay across websites.
- In the development of a scalable web application, we employ a number of technologies to maintain the quality of the service.
- *IaaS* reduces efforts that come with the manual administration of physical computing infrastructure.
- Testing tools applied to the platform (*Apache JMeter* y *SeleniumHQ*) give a global vision of its performance and functionality. They provide the means to reproduce the stress and usability tests.

ACKNOWLEDGMENT

The authors want to thank all the people that made this project possible, especially their families and colleagues.

Special thanks go to Raquel Shida, Mathias André and Lars Lindberg Christensen from the European Southern Observatory to trust the capabilities of the proposed solution. Jerson Pabón from Mobile Corp S.A.S and Camilo Riveros from Parquesoft Meta for logistic support. Juan Vargas collaborated with the translation of the document.

DISCLAIMER

Some of the algorithms implemented in this solution are under Provisional Application by the United States Patent and Trade Mark Office (USPTO), Application Number 62582968.

REFERENCES

[1] "9 herramientas para comprobar enlaces rotos en una web | Programando mi web," *programandomiweb*, 2016. [Online]. Available: http://www.programandomiweb.com/herramientas-enlaces-rotos-pagina-web/. [Accessed: 22-Jun-2017].

[2] Microsoft, "¿ QUÉ ES IAAS ? Infraestructura como Servicio," *Microsoft Azure*. [Online]. Available: https://azure.microsoft.com/es-es/overview/what-is-iaas/. [Accessed: 23-May-2017].

[3] Amazon Web Services Inc, "¿Qué es AWS? - Amazon Web Services," *amazon.com*, 2017. [Online]. Available: https://aws.amazon.com/es/what-is-aws/. [Accessed: 04-Jun-2017].

[4] A. W. S. Inc, "AWS | Elastic beanstalk para aplicaciones web desarrolladas con Java," *amazon.com*, 2017. [Online]. Available: https://aws.amazon.com/es/elasticbeanstalk/. [Accessed: 11-Jun-2017].

[5] A. W. S. Inc, "AWS | Almacenamiento de datos seguro en la nube (S3)," *amazon.com*, 2017. [Online]. Available: https://aws.amazon.com/es/s3/. [Accessed: 06-Oct-2017].

[6] I. Amazon Web Services, "Amazon RDS para PostgreSQL," *amazon.com*, 2015. [Online]. Available: https://aws.amazon.com/es/rds/postgresql/. [Accessed: 08-Jun-2017].

[7] A. W. S. Inc, "AWS | Elastic compute cloud (EC2) de capacidad modificable en la nube," *amazon.com*, 2017. [Online]. Available: https://aws.amazon.com/es/ec2/. [Accessed: 06-Dec-2017].

[8] Amazon Web Services Inc, "AWS | Amazon Simple Email Service (SES) - Cloud Based Email Services," *amazon.com*, 2017. [Online]. Available: https://aws.amazon.com/es/ses/faqs/. [Accessed: 08-Jun-2017].

[9] A. W. S. Inc, "DNS de nube administrado – Sistema de nombres de dominio – Amazon Route 53 | AWS," *amazon.com*, 2017. [Online]. Available: https://aws.amazon.com/es/route53/. [Accessed: 06-Oct-2017].

[10] Koehler, Wallace. "A longitudinal study of Web pages continued: a consideration of document persistence". Information Research. Archived from the original on 11 September 2017. Retrieved 16 October 2017.