International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 4, Number 17 (2014), pp. 1783-1789

http://www. irphouse.com

Framework for Intelligent Crawler Engine on IaaS Cloud Service Model

Pratibha Ganapati Gaonkar[1], Dr. Nirmala C R[2]

M. Tech (Dept. of CS&E),
Bapuji Institute of Engineering and Technology, Davangere, Karnataka, India.
Professor & HOD (Dept. of CS&E),
Bapuji Institute of Engineering and Technology, Davangere, Karnataka, India.
prathibha. kwr@gmail. com, crn@bietdvg. edu

Abstract

This paper is aimed to implement an intelligent crawler engine on cloud computing framework. This approach uses virtual machines on a cloud computing framework to run intelligent crawler engine. The use of Virtual Machine (VM) on this framework will help for easy setup/installation, maintenance or VM terminating that has been running with some particular crawler engine as needed. With this Framework, we have designed an intelligent crawler by making use of Naive Best First algorithm and R-Spam Rank algorithm, which is more efficient compared to the earlier crawlers as per the result and analysis. In order to accomplish this task Amazon public cloud is used with its services, S3 and EC2.

Keywords: Cloud computing Framework; Elastic Compute Cloud (EC2); Intelligent Crawler engine; R-SpamRank; Simple Storage Service (S3).

Introduction

In the world of Web 2. 0, the adage "content is king" remains a prevailing theme. With seemingly endless content available online, the "findability" of content becomes a key factor. Search engines are the primary tools people use to find information on the web. Searches are performed using keywords. Web crawlers are the programs or software that uses the graphical structure of the Web to move from page to page [1]. Such programs are also called wanderers, robots, spiders, and worms. Web crawlers are designed to retrieve Web pages and add them or their representations to local repository/databases. Web crawlers are mainly used to create a copy of all the visited pages for later processing by a search engine that will index the downloaded

pages that will help in fast searches. Web search engines work by storing information about many web pages, which they retrieve from the WWW itself.

A crawler for a large search engine has to address two issues. First, it has to have a good crawling strategy, i. e., a strategy for deciding which pages to download next. Second, it needs to have a highly optimized system architecture that can download a large number of pages per second while being robust against crashes, manageable, and considerate of resources and web servers [2] [3]. Topical crawlers or focused crawlers were developed to create contextual search engine or more focused result [6] [14]. Topical crawlers have been used in a variety of applications such as competitive intelligence search engines and digital libraries. The Naive Best-First crawler can be used for crawling the web [7].

Cloud computing can really be a winsome option for an enterprise. Especially for the new enterprises, which want to reduce the upfront cost for their computing infrastructure. Even established organizations can reduce not only the computing infrastructure cost, but also the administrative and operational cost for the infrastructure. Because after purchasing the computing infrastructure, the organization needs human resources, space, energy and many other resources to manage and administer them. Whereas, in the case of opting for cloud computing services, these costs are reduced [4]. Some of the cloud infrastructure/service providers are Amazon [5], Salesforce, Google App Engine and Microsoft Azure.

Spam web pages intend to achieve higher-than-deserved ranking by various techniques. While human experts could easily identify spam web pages, the manual evaluating process of a large number of pages is still time consuming and cost consuming. The R-SpamRank algorithm can be used for detection of spam pages [8].

"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e. g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [9]. There are three main categories of service models of cloud computing: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [10]. Scalability is one of the most prominent characteristics of all three categories [11] [12]. The IaaS systems can offer elastic computing resources like Amazon Elastic Compute (EC2) and on demand storage resources like Amazon's Simple Storage Service (S3). Two of the most common deployment models of cloud computing are public cloud infrastructure and private cloud infrastructure. The former are the cloud computing infrastructure provided by 3rd party service provider (such as Google and Amazon) based on pay-as.-you-use model and the latter is the cloud computing infrastructure set up and managed by an organization for its own use.

Key advantages of cloud computing is the use of virtualization so that the users do not need to know where the computation performed by a machine. Also, with the usage of VM (s) will make it easier when running application and operating system installation. By using the VM, we can easily create a master application or service built in an operating system resulting in image. If we need some similar system to run the same program then cloud computing will easily turn on or duplicate the same VM on a particular physical node computer without hardware installation [13].

Proposed Architecture

This paper tries to build an intelligent crawler engine service on the cloud computing framework. The main objectives of this paper are:

- To design an intelligent crawler engine on cloud computing framework by making use of Naive Best First algorithm and R-SpamRank algorithms.
- To design the module to save keyword and for more effective searching.
- To create specific buckets in cloud storage to save, record and indexing the results from crawler engine.

The proposed framework of crawler engine in the figure 1 consists of virtual machines running on EC2 instances with crawler engine services and application servers and storing the query based results of crawling on S3. The crawler uses Naive Best-First crawling strategy.

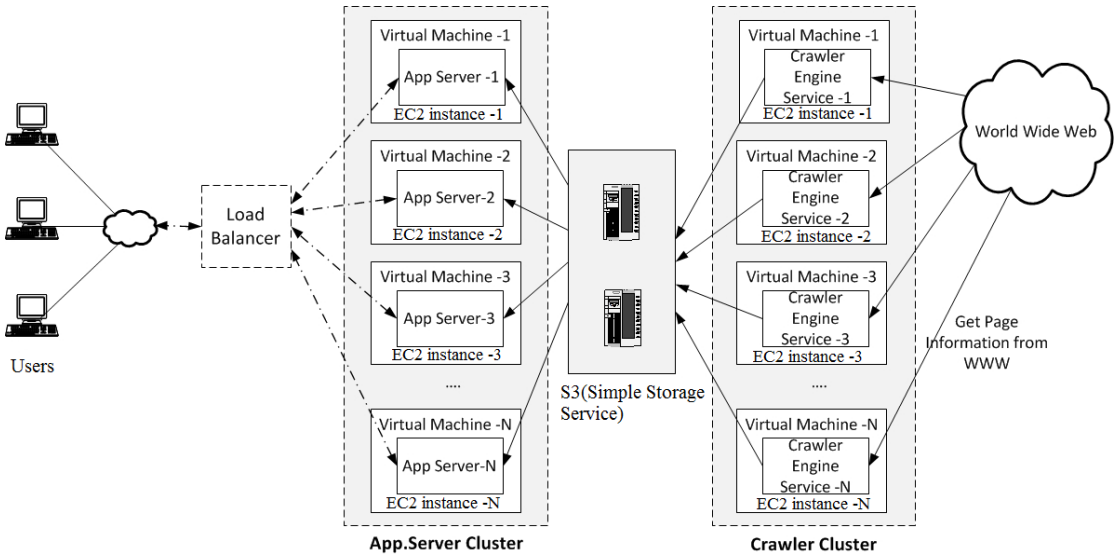

Figure 1: Framework for Crawler Engine on Cloud.

The Naive Best-First crawler represents a fetched Web page as a vector of words weighted by occurrence frequency. The crawler then computes the cosine similarity of the page to the query or description provided by the user, and scores the unvisited URLs on the page by this similarity value. The URLs are then added to a frontier that is maintained as a priority queue based on these scores. In the next iteration each crawler thread picks the best URL in the frontier to crawl, and returns with new unvisited URLs that are again inserted in the priority queue after being scored based on the cosine similarity of the parent page. The cosine similarity between the page p and a query q is computed analogous to the equation (1).

$$sim(p,q) = \frac{V_p . V_q}{\|V_p\| . \|V_q\|} \quad (1)$$

Where vq and vp are term frequency (TF) based vector representations of the query and the page respectively.

Despite extensive research, spam filtering techniques generally fall short for protecting the web services. To better address this need, we present R-SpamRank, that crawls URLs as they are submitted to web services and determines whether the URLs direct to spam and helps the crawler whether to proceed with crawling or not for that URL. R-SpamRank algorithm aims to detect spam web pages. In this algorithm, the web page gains the spam rank value through forward links, which are the links of reverse direction used in traditional link-based algorithm. Therefore, this algorithm is called as R-SpamRank which means reverse spam rank.

This algorithm uses a blacklist containing spam web pages as seeds. The blacklist is manually collected in the experimental system. We assigned an initial R-SpamRank value for each page in the blacklist, and these values would expand in the iterative computation to the web pages linking to them. The formula of the algorithm is shown in equation (2) below.

$$RSR(A) = (1-\lambda)I(A) + \lambda \sum_{i=1}^{n} \frac{RSR(T_i)}{C(T_i)} \quad (2)$$

$$I(A) = \begin{cases} 1 \; if \; A \; in \; blacklist \\ 0 \; otherwise \end{cases} \quad (3)$$

where *RSR (A)* is the R-SpamRank value of page A; λ is a damping factor, which is usually set to 0. 85; I (A) is the initial value for page A, it is set to 1 if page A in the original blacklist, otherwise 0 as shown in (3); n is the number of forward links of page A, and *Ti* is the ith forward link page of page A; C (*Ti*)is the number of in links of Page *Ti*; *RSR (Ti)* is the R-SpamRank value of page *Ti*.

Experimental Results

In order to run service engine crawlers on cloud framework the instances of amazon EC2 are launched, to use the virtual machines to run the crawler engines. Amazon Elastic Compute Cloud (Amazon EC2) provides resizable computing capacity in the Amazon Web Services (AWS) cloud. After the instance is been launched and connected by using the key pair, the crawler engine services are run on the virtual machines. Then the indexed score values of crawling are stored in the Amazon Simple Storage Service (Amazon S3). We can use Amazon S3 to store and retrieve any amount of data at any time, from anywhere on the web. We accomplish these tasks using the simple and intuitive web interface of the AWS Management Console. In figure 2 we have plotted graph for crawled pages for Jobs domain by taking domain versus processed pages for two minutes. In the graph for the domain of Jobs (Freshersworld, Shine, Monster, Fresherslive) the processed pages are 189, 268, 55 and 73 respectively.

In figure 3 we have plotted graph for crawled pages for keywords in domain by taking domain keywords versus processed pages for two minutes. For the keyword Govt the jobs domain (Freshersworld, Shine, Monster, Fresherslive) the processed pages are (34, 214, 9, 2). For the keyword Engineering the jobs domain (Freshersworld, Shine, Monster, Fresherslive) the processed pages are (109, 219, 17,

11). For the keyword IT the jobs domain (Freshersworld, Shine, Monster, Fresherslive) the processed pages are (171, 221, 54, 16). For the keyword Research the jobs domain (Freshersworld, Shine, Monster, Fresherslive) the processed pages are (33, 212, 53, 21).

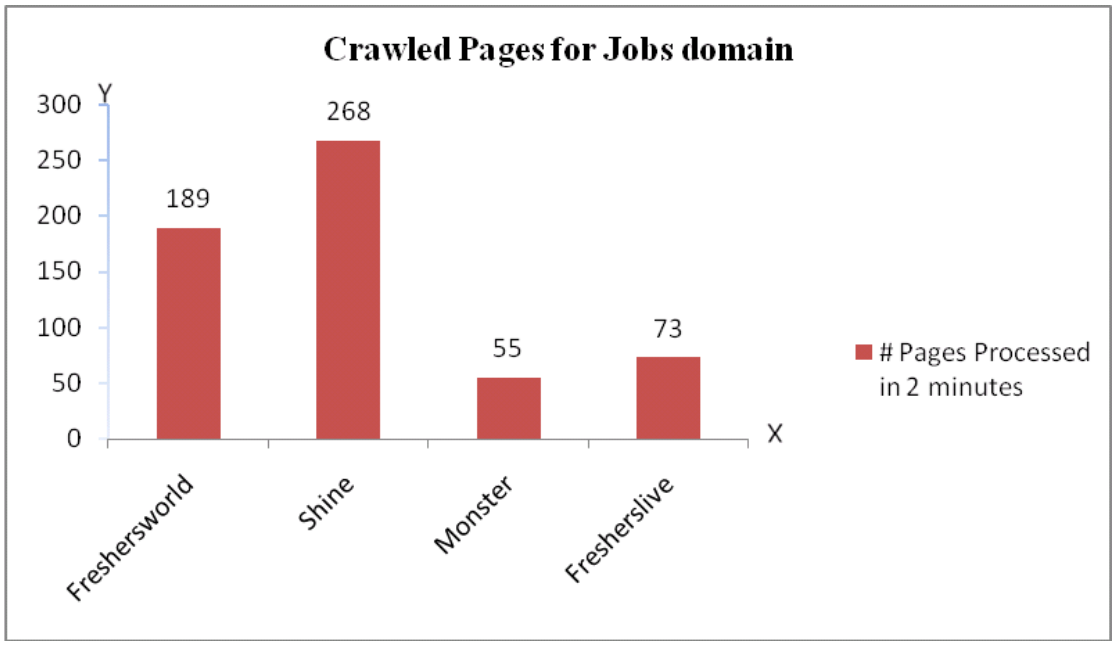

Figure 2: Crawled Pages for Jobs domain. X: domain. Y: pages processed.

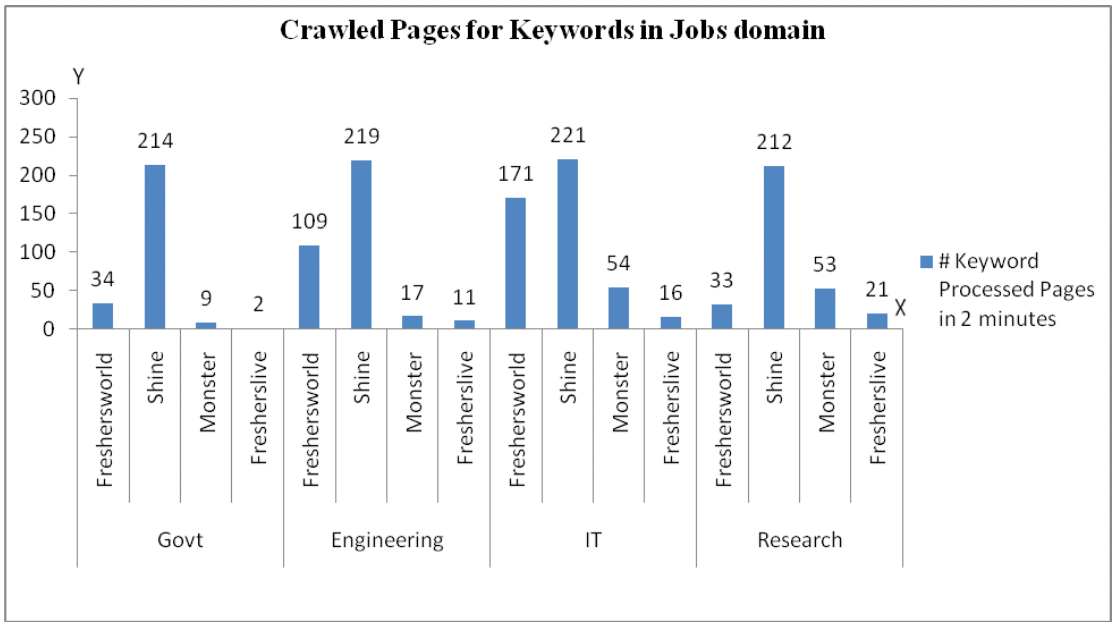

Figure 3: Crawled Pages for Keywords in Jobs domain. X: domain keywords. Y: pages processed

By using Amazon EC2 virtual machine, we have implemented intelligent crawler engine on cloud computing framework using the algorithms Naive Best-First and R-SpamRank. The intelligent crawler sends the url streams to R-SpamRank. Based on its decisions the spam free urls are crawled and based on this crawling we have analyzed the graphs in figure 2 and figure 3 and conclude that the jobs domain Shine is more popular than the other three jobs domain Freshersworld, Monster and Fresherslive.

Conclusion

In this paper we have implemented intelligent crawler engine on cloud computing framework. This approach uses virtual machines on EC2 to run crawler engine services and stores the crawled indexed results on S3. We have conducted experiments for Jobs domain like Freshersworld, Shine, Monster and Fresherslive with keywords Govt, Engineering, IT and Research. Based on experiments conducted we conclude that the jobs domain Shine is more popular than the other three jobs domain Freshersworld, Monster and Fresherslive. With the huge data processed on the cloud, the Intelligent crawler engine designed, efficiently crawls, processes and stores the results on S3.

References

[1] GautamPant, Padmini Srinivasan, and FilippoMenczer: “Crawling the Web”.

[2] J. Cho, H. Garcia-Molina, and L. Page. Efficient crawling through url ordering. In *7th Int. World Wide Web Conference*, May 1998.

[3] M. Najork and J. Wiener. Breadth-first search crawling yields high-quality pages. In*10th Int. World Wide Web Conference*, 2001.

[4] J. Hurwitz, R. Bloor, M. Kaufman, F. Halper, Cloud Computing for Dummies, Wiley Publishing, Inc. 2009.

[5] Amazon Elastic Compute Cloud, https: //aws. amazon. com/ec2/

[6] Gautam Pant and Padmini Srinivasan, “Link Contexts in Classifier-Guided Topical Crawlers”, *IEEE Transactions On Knowledge and DataEngineering, Vol. 18, No. 1, January 2006*.

[7] C. Olston and E. H. Chi, “ScentTrails: Integrating Browsing and Searching on the Web”, ACM Trans. Computer-Human Interaction, vol. 10, no. 3, pp. 177-197, Sept. 2003.

[8] Chenmin Liang1, Liyun Ru2 and Xiaoyan Zhu1, “R-SpamRank: A Spam Detection Algorithm Based on Link Analysis”.

[9] P. Mell and T. Grance, "The NIST Definition of Cloud Computing, " National Institute of Standards and Technology, USA2009.

[10] A. Lenk, *et al.*, "What is Inside the Cloud? An Architectural Map of the Cloud Landscape, " presented at the Workshop on Software Engineering Challenges of Cloud Computing, Collocated with ICSE 2009 Vancouver, Canada, 2009.

[11] R. Grossman, "The Case for Cloud Computing, " *IEEE Computer,* vol. 11, pp. 23-27, 2009.
[12] Q. Zhang, *et al.,* "Cloud computing: state-of-the-art and research challenges, " *Journal of Internet Services and Applications,* vol. 1, pp. 7-18, 2010.
[13] Sinung Suakanto, Suhono H. Supangkat, Suhardi, Roberd Saragih, " Building Crawler Engine On Cloud Computing Infrastructure ".
[14] Soumen Chakrabarti, Martin van den Berg, Byron Domc. "*Focused crawling: a new approach to topic-specific Web resource discovery*", 1999.
[15] Amazon Simple Storage Service, https: //aws. amazon. com/s3/