

Design and Implementation of the Hadoop-based Crawler for SaaS Service Discovery

Asma Alkalbani¹, Akshatha Shenoy¹, Farookh Khadeer Hussain², Omar Khadeer Hussain³, Yanping Xiang⁴

^{1,2} Decision Support and e-Service Intelligence Lab

Quantum Computation and Intelligent Systems,

School of Software, University of Technology

Sydney, NSW 2007, Australia

{Asma.M.Aikalbani, Akshatha.Shenoy}@student.uts.edu.au; Farookh.Hussain@uts.edu.au

³ School of Business

University of New South Wales Canberra (UNSW Canberra)

Australian Defence Force Academy

Canberra, ACT, 2602

O.Hussain@adfa.edu.au

⁴ Collaborative Autonomic Computing Lab

University of Electronic Science & Technology, Chengdu, China

Abstract - Software as a Service is the most adopted cloud service (46%) compared with Infrastructure as a Service (IaaS) (35%) and Platform as a Service (PaaS) (34%) [1]. Currently, the capability of discovering a SaaS of interest online across multiple cloud providers and reviews websites is a significant challenge, especially when using general search mechanisms (Google and Yahoo!) and search tools provided by existing reviews and directories. Discovering a SaaS is time-consuming, requiring consumers to browse several websites to select the appropriate service. This paper addresses the issues related to the efficient discovery of SaaS across review websites by developing the SaaS Nutch Hadoop-based Crawler Engine - SaaS Nbased Crawler. The crawler is capable of crawling cloud reviews to find SaaSs of interest and enable the establishment of a central repository that could be used to discover SaaSs much more efficiently. The results show that the SaaS Nbased crawler can effectively crawl review websites and provide a list of the latest SaaS being offered.

Keywords - *Software as Service, Service Discovery, Nutch, Hadoop based crawler, SaaS Repository.*

I. INTRODUCTION

Cloud computing is a new computing paradigm where computing resources, such as hardware, software development platforms and software applications are offered “as-a-service” to end-users [2]. Software as a Service (SaaS) refers to software applications owned and provisioned by service providers to consumers on demand [2]. A recent report by Gartner [3] points to a healthy and steady growth of SaaS adoption from 2013 to 2016.

Despite the high adoption of SaaS, finding and selecting an appropriate SaaS remains an outstanding research issue [4]. In the past few years, there has been a

lot of research focus on proposing and developing intelligent methods for SaaS selection. These methods assume the availability of information by SaaS providers that match end-users’ search queries or requirements, and focus on developing methods to select the SaaS that best matches a user’s requirements. However, they do not focus on methods for discovering SaaS or building a repository of SaaS services. Recently, a number of cloud review websites and directories (such as CloudReviews¹ and GetApp²) have appeared that provide a listing of cloud services. These sites usually collect service information from the cloud providers’ official websites and present them via a single portal. Information, such as customer reviews of cloud providers is provided which may assist consumers in searching for and comparing services. Although the cloud review websites could be useful to consumers in selecting a provider, these websites do not provide up-to-date service information. As a result, these websites may contain unreliable service information, for instance the service price, and the Quality of Service (QoS) at the time the consumers are making a service selection decision.

The lack of a standard representation format to represent cloud services across different publishing platforms, such as cloud review websites, compounds the issue of cloud service discovery [5]. Due to the increase in the number of cloud providers compared to many similar cloud services (such as storage services), decision making

either along a single or multiple dimensions is a time-consuming process.

Crawlers are an important component of search engines. Studies point out that around 85% of Internet users use search engines to find information from the WWW [6]. Making use of general purpose search engines (such as Google, Yahoo, Bing etc...) for searching for cloud services may result in imprecise and irrelevant search results with irrelevant information being retrieved. However, in the existing literature there is limited work on proposing and developing a focused crawler for crawling cloud services. Some researchers have proposed the use of existing generic search engines coupled with a cloud service ontology to enhance the search for cloud services [7, 8]. Others [9] focus on developing an overall methodology and framework for a cloud service discovery system. A critical shortcoming in their method is that the service information has to be performed manually [9]. To the best of our know¹<http://www.cloudreviews.com> used the use of a crawler²<http://www.getapp.com/> or et al. [10] developed a general crawler for cloud service discovery to establish a cloud service dataset that could provide useful information on cloud services. The primary limitation of their work is that the cloud service information in the repository could become out-of-date easily and is not up-to-date. As a result, there is a need to develop a dynamic mechanism to update the information in the repository. To address this significant limitation of the existing literature on the cloud service discovery, we propose the *SaaS Nhbse crawler*. This crawler is based on an open source general crawler that uses the Hadoop framework to crawl and process information from multiple cloud services simultaneously. Additionally, the proposed SaaS NHBse crawler makes use of Apache Solr as a query interface. The results of this focused crawling are stored in a central repository for public use. To the best of our knowledge, this study is the first to use the Apache Nutch crawler with Apache Solr for cloud service discovery. More specifically, the contributions of our work are as follows:

- This study proposes the use of the Apache Nutch-based crawler and Apache Solr integration for cloud service discovery;
- As a result of the crawling process, this study provides cloud consumers and providers with the first reliable SaaS public repository. Both consumers and providers may use this public repository for various activities such as service advertisement, searching, composition etc.

The rest of this paper is structured as follows. In Section II, we discuss the related work. In Section III, we discuss the advantages of making use of a Hadoop-based architecture for the focused crawling of cloud services. In Section IV, we present an overview and the architecture of the proposed SaaS Nhbse crawler. In Section V, we outline the results obtained from the crawling and discuss them. Finally, Section VI concludes this work and discusses aspects for further research.

II. RELATED WORK

In this section, we briefly introduce previous work in the area of cloud service discovery and also briefly mention the technological underpinnings of the proposed methods. Service discovery for various paradigms, such as web services and online services, and in various application domains, such as the transportation domain etc. has attracted a lot of research attention in the past few years, resulting in several methods that improve service discovery [11, 12]. A discussion on this work is outside the scope of this research. Within the area of cloud service discovery, most of the existing research studies focus on the use of domain ontologies to enhance service discovery. Kang et al. [8] propose the Cloudle system which uses existing general purpose search engines (Google, Yahoo!) to collect service information based on a user's query and then finds the most relevant page by consulting ontology concepts. In other work, Afify et al. [9] proposed a semantics-based approach to enhance the process of searching for and selecting SaaS. They proposed a unified SaaS ontology to help in the SaaS search process. Also, they use ontology concepts to structure the system's repository. In their approach, they used several parameters, such as service characteristics and QoS in selecting a SaaS. A key shortcoming of their work is that each cloud provider needs to register its details in the system and manually enter their own cloud service information, for instance, service type, QoS, price etc.

Crawlers are widely used for indexing and to assist in the process of searching for web resources. In the past few years, a lot of research attention has been devoted to this area with a view to developing domain-specific, purpose-specific crawlers [13-15]. As mentioned above, ontologies have been widely used as an enabling technology for cloud service discovery. However, there is little existing research on proposing and developing an intelligent crawler to *intelligently* crawl and index cloud service information, along with its attributes (such as QoS, price etc.), in an *efficient* manner. Noor et al. [10]

proposed and developed a crawler engine that is able to crawl web portals (Yahoo, Google) using a cloud ontology and then generate cloud service information. Their work is the first proposed work in the literature on developing and defining crawlers that are able to crawl cloud service information. However, the collected dataset on cloud services has several limitations such as: 1) the dataset is incomplete, in the sense that it lacks key service information (such as service name and URL); 2) the data values in the dataset do not have the corresponding semantics associated with it. As a result, it is difficult for cloud consumers to understand its meaning for decision making.

We aim to build on the work of Noor et al. [10] to enhance their approach by addressing the aforesaid limitations. We additionally plan to propose efficient methods to crawl and index cloud services. By efficient, we mean that we will develop distributed methods for cloud service crawling. In this paper, we propose the use of the Apache Nutch crawler (based on the Hadoop framework) with Apache Solr to enhance the cloud service search. Additionally, as a result of the crawling process, we aim to establish a publicly available central service repository. Apache Nutch has been used as a specific domain crawler to improve the search for information in different fields [16, 17], however, to the best of our knowledge, the use of Apache Nutch has not

previously been investigated to address the service discovery issue. Our study is the first work that uses the Apache Nutch– Hadoop based crawler with Solr integration as a query interface.

III. WHY HADOOP

Apache Hadoop is a highly scalable framework that helps in the distributed processing of a large volume of data. **When a single computer node is not able to handle large volumes of data, in the Hadoop framework, the task is carried out by breaking the data down to smaller sections by creating one or more shards and assigning each shard to one processing or computer node.** In the Hadoop framework, new nodes can be added at any point in time, depending on the need and the volume of data to be processed. In addition, the Hadoop framework has a load balancer that helps in maintaining the load equally between servers. Moreover, it is a schema-less database which can handle data in any format such as structured, unstructured, images, audio files etc. This is a huge advantage in web crawling as the crawler is able to crawl a web resource, without being specific or limited to the data that needs to be crawled. Figure 1 shows the overall architecture of the proposed SaaS Hhbase Crawler.

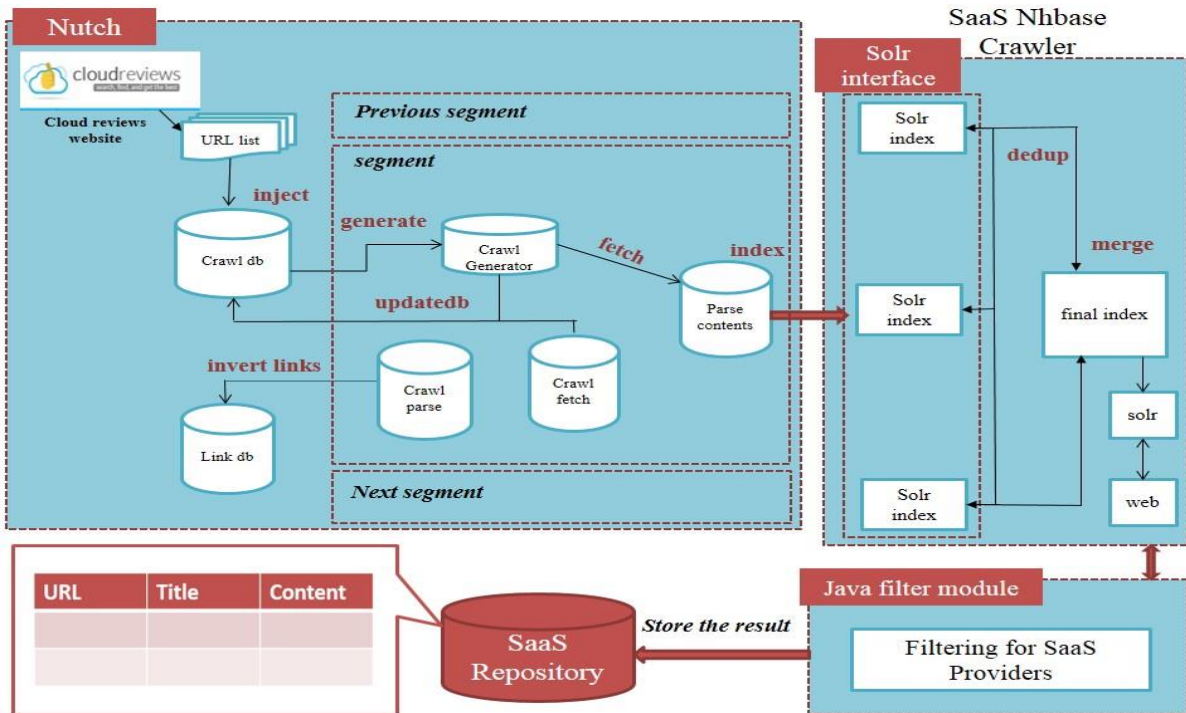


Figure 1. System architecture and workflow of the proposed SaaS Hhbase Crawler

IV. OVERVIEW OF SAAS NBASE CRAWLER

A. System Architecture:

As presented in Figure 1, the overall architecture of the system comprises four components: the Nutch crawler, the Solr interface, the Java filter module and the SaaS repository. We now briefly introduce the function of each.

- a) Nutch Crawler: Apache Nutch is an open source web crawler, written in Java. Its framework facilitates customized crawling by retrieving, parsing and indexing websites based on our requirements. It can run on a distributed environment, which utilizes the Hadoop framework for the parallel processing of large-scale web crawling. This helps to expedite the crawling and parsing process. Apache Nutch utilizes HBase as its data store to store a large volume of data. Apache Nutch can be easily integrated with Apache Solr which we use in our framework to index the crawled web pages.
- b) Solr interface: Apache Solr is a very powerful, highly scalable tool which facilitates a distributed search by indexing the crawled information. Also, Apache Solr can be queried to filter the results.
- c) Java filter module: The role of this module is to filter the crawling results to obtain only the SaaS providers' websites. Once all the fetched URLs from Apache Nutch are inserted into the Apache Solr, the next process is to filter the URLs based on the filter criteria. This module filters all the indexed URLs from Apache Solr to identify only SaaS URLs. We have listed several words to identify URLs as SaaS sites. We also make use of WordNet® to assist in providing the related synonyms to identify only the SaaS sites from a list of URLs.
- d) SaaS Repository: The final step is to develop a SaaS repository which holds only the SaaS providers mentioned on the CloudReviews.com website. This is used to store information on the crawled SaaS with its characteristics.

B. System Workflow:

In this section, we introduce the step-by-step working of the crawler.

a) Apache Nutch working process

Step 1: As shown in Figure 1, Apache Nutch crawling starts with Seed.text. The Seed.text file contains all the URLs which need to be crawled.

Step 2: In the Apache Nutch framework, CrawlDB will encompass all the URLs that need to be crawled.

Step 3: In the crawler, the job is to retrieve (in segments) the URLs to be crawled.

The retrieval function can retrieve URLs to a depth of 'n' level. In our implementation of the crawler, we retrieved URLs up to 3 levels deep to obtain relevant information. It retrieves a large number of URLs for each iteration and breaks these URLs down into different segments. These segments are run on different processors for parallel processing. As shown in Figure 1, the retrieved segments of the URLs are run on different servers for the parallel processing of a large volume of data.

Step 4: Once the URLs have been crawled, the parser is invoked to parse all the crawled information. It parses both text and metadata.

Step 5: The Hbase is updated with the results from the retrieved job. This updates all the retrieved URLs as well as the last retrieved URL from the retrieve job. This is an iterative process up to 'n' iterations, depending on the depth specified. Then the database has all the retrieved URLs along with their status, such as retrieved or not retrieved.

Step 6: The inverted links job creates a web graph of all the retrieved URLs and stores this in the Link DB. This helps in the indexing of incoming anchor text with the pages.

Step 7: Once all the URLs have been crawled by Apache Nutch, Apache Solr indexes all the URLs that have been crawled.

b) Solr interface working process

Apache Solr is an open source software, written in Java. Solr is a very powerful, highly scalable tool which facilitates distributed search and index replication. The working process of SOLR in our crawler is as follows:

Step 1: The URLs retrieved by Apache Nutch are injected into Apache Solr.

Step 2: The Apache Solr indexer indexes these URLs using a unique identifier.

c) Java filter working process:

The filter is invoked when all the URLs have been parsed and indexed. As previously mentioned, this module filters all indexed URLs in the Apache Solr to identify only the SaaS providers' URLs. The next step is to retrieve the Apache Solr index by passing our query based on the criteria. The criteria are a list of SaaS keywords that identify URLs as SaaS sites. Additionally, we make use of WordNet® that assists in providing related synonyms to identify only SaaS. The step-by-step working of the Java filter is as follows:

Step 1: Establish a connection with the Solr server.

Step 2: Specify a query to retrieve content. We used WordNet® which is a database in the English language that groups English words into sets of synonyms. In the query, we specified the filter conditions to filter only SaaS provider websites. The filter condition contains SaaS and SaaS sets of synonyms.

Step 3: Pass this query to the Solr server to retrieve the filtered result. The results are the URLs of the SaaS, the title of the SaaS and all the contents of the SaaS provider's website.

d) SaaS Repository:

Finally, the output from the Apache Solr query is inserted into the SQL database (which currently serves as the repository). To do this, a connection is established with the SQL database. Then, the output from the Solr query is inserted into the table which forms the SaaS repository. The repository contains all the information from the SaaS providers from the CloudReview website.

V. RESULT & DISCUSSION

In order to run the process documented in Section IV, once all the settings are complete, we go to the local directory of Apache Nutch (from the terminal) and run the following commands:

Step 1: Start the Apache server using the command `java -jar start.jar`

Step 2: Start Hbase using the command `./bin/start-hbase.sh`

Step 3: Start web crawling using the command
`bin/crawl urls/seed.txt TestCarwl`
`http://localhost:8983/solr 2`

The last command takes the URL links as input which are provided in the file `seed.txt`. When Apache Nutch starts crawling, it reads the file `seed.txt` file and takes all the URLs specified in this file, as input. For our research, we provide `www.cloudreview.com` as an input to Apache Nutch.

Cloud Review provides unbiased reviews on cloud hosting, managed cloud hosting, cloud storage etc. [19, 20]. It provides basic information on service providers, the ratings of service providers (by their customers), and a comparison of similar service providers in the market. Finally, it provides a corresponding link to the cloud provider's website. It is almost like a one-stop shop for information about all the cloud service providers. In our work, we crawl the service provider's webpage to ensure greater accuracy of the provider's information. The advantage of crawling the provider's webpage is that all the information on their website is up-to-date.. Using our filter condition, we filter only the service provider's website. Figure 2 shows all the retrieved URLs parsed by Apache Nutch. During the first iteration, it retrieves all the URLs up to three levels of depth. Once Apache Nutch retrieves and parses all the URLs from the Cloudreviews website up to a depth of 3, the information is indexed on Apache Solr

Next, we use filter keywords such as *Software, program, Computer software, Software systems, software packages, and packages* to flag URLs as SaaS websites.

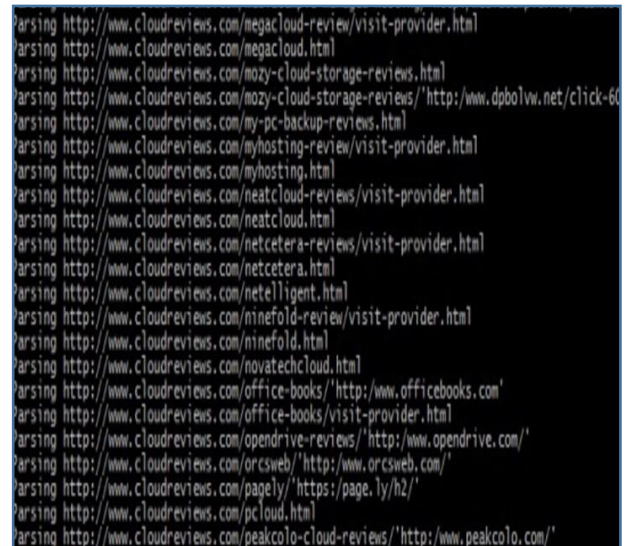


Figure 2. The URL Retrieval & Parsing Process from Nutch

Figure 3 shows a snapshot of the SaaS repository. The repository displays SaaS information in three columns, the *SaaS provider's URL*, *SaaS Service Name*, and *Content*. The content column has information such as price, quality of service and service rating.

ID	URL	Title	Content
1	http://www.cloudreviews.com/hyre-managed-hosting...	Hyre Hyre Review Hyre Managed Hosting	Hyre Hyre Review Hyre Managed Hosting News Boost Your Business with Innov
2	http://www.cloudreviews.com/acronis.html	Acronis Acronis Review Acronis Online Backup	Acronis Acronis Review Acronis Online Backup Provider News Boost Your Business
3	http://www.cloudreviews.com/vero.html	Vero Cloud Vero Reviews Vero Cloud Hosting	Vero Cloud Vero Reviews Vero Cloud Hosting Service News Boost Your Business
4	http://www.cloudreviews.com/digitalocean.html	DigitalOcean DigitalOcean Review DigitalOcean	DigitalOcean DigitalOcean Review DigitalOcean Cloud Hosting News Boost Your B

Figure 3. Snapshot of Crawling and result.

VI. CONCLUSION & FUTURE WORK

In this paper, we proposed the use of focused crawlers for crawling and indexing SaaS. The overall architecture and working of the SaaS crawler was outlined and discussed. It comprises four major components and the working of each component was discussed. The proposed crawler design involves the use of open source Apache Nutch crawling. It additionally makes use of Apache Solr for indexing the crawler information. As part of this crawler, we developed a Java filtering module to filter the result and only obtain links for SaaS providers. The result of the overall crawling process is stored in a SaaS repository as a three-tuple.

For future work, we plan to extend our work by developing the SaaS Semantic Crawler Engine that has the ability to crawl for service discovery more efficiently. We additionally plan to crawl more than one cloud review site and compare the results obtained by crawling multiple cloud review websites.

VII. Acknowledgement

This work is partly supported by NSFC grant number 61350110517.

REFERENCES

- <http://www.kpmg.com/Global/en/IssuesAndInsights/ArticlesPublications/Documents/cloud-clarity.pdf> (Accessed 2nd November 2014)
- Mell, P., and Grance, T., 'The NIST definition of cloud computing', 2011
- Gartner: 'Forecast Overview: Public Cloud Services, Worldwide, 2011-2016, 2Q12 Update', in Editor (Ed.): 'Book Forecast Overview: Public Cloud Services, Worldwide, 2011-2016, 2Q12 Update' (2012, edn.), pp. 18-19
- http://ec.europa.eu/digital-agenda/events/cf/e2wp2014/document.cfm?doc_id=23902. (Accessed 18th November 2014)
- Gracia, J., and Mena, E., 'Semantic heterogeneity issues on the web', Internet Computing, IEEE, 2012, 16, (5), pp. 60-67
- Kobayashi, M., and Takeda, K., 'Information retrieval on the web', ACM Computing Surveys (CSUR), 2000, 32, (2), pp. 144-173
- Nagireddi, V.S.K., and Mishra, S., 'An ontology based cloud service generic search engine', in Editor (Eds.): 'Book An ontology based cloud service generic search engine' (IEEE, 2013, edn.), pp. 335-340
- Kang, J., and Sim, K.M., 'Cloudle: An Agent-based Cloud Search Engine that Consults a Cloud Ontology', in Editor (Eds.): 'Book Cloudle: An Agent-based Cloud Search Engine that Consults a Cloud Ontology' (2010, edn.), pp. 312-318
- Afify, Y.M., Moawad, I.F., Badr, N.L., and Tolba, M.F., 'A semantic-based Software-as-a-Service (SaaS) discovery and selection system', in Editor (Eds.): 'Book A semantic-based Software-as-a-Service (SaaS) discovery and selection system' (2013, edn.), pp. 57-63
- Noor, T.H., Sheng, Q.Z., Alfazi, A., Ngu, A.H.H., and Law, J., 'CSCE: A Crawler Engine for Cloud Services Discovery on the World Wide Web', in Editor (Eds.): 'Book CSCE: A Crawler Engine for Cloud Services Discovery on the World Wide Web' (2013, edn.), pp. 443-450
- Al-Masri, E., and Mahmoud, Q.H., 'Investigating web services on the world wide web', in Editor (Eds.): 'Book Investigating web services on the world wide web' (ACM, 2008, edn.), pp. 795-804
- Dong, H., Hussain, F.K., and Chang, E., 'Ontology-learning-based focused crawling for online service advertising information discovery and classification': 'Service-Oriented Computing' (Springer, 2012), pp. 591-598
- Ukhopadhyay, D., Biswas, A., and Sinha, S., 'A new approach to design domain specific ontology based web crawler', in Editor (Eds.): 'Book A new approach to design domain specific ontology based web crawler' (IEEE, 2007, edn.), pp. 289-291
- McCallumzy, A., Nigamy, K., Renniey, J., and Seymorey, K., 'Building domain-specific search engines with machine learning techniques', 1999
- Chakrabarti, S., Van den Berg, M., and Dom, B., 'Focused crawling: a new approach to topic-specific Web resource discovery', Computer Networks, 1999, 31, (11), pp. 1623-1640
- Medelyan, O., Schulz, S., Paetzold, J., Poprat, M., and Markó, K., 'Language specific and topic focused web crawling', in Editor (Eds.): 'Book Language specific and topic focused web crawling' (2006, edn.), pp. 865-868
- Chan, S.-B., and Yamana, H., 'The method of improving the specific language focused crawler', in Editor (Eds.): 'Book The method of improving the specific language focused crawler' (2010, edn.), pp. 699-707.
- Shaikh, A., and Laliwala, D.Z.: 'Web Crawling and Data Mining with Apache Nutch', Packt Publishing, 2013.
- <http://www.cloudreviews.com/>
- <http://www.getapp.com/>