JAICT
ISSN 2355 1771
EISSN 2723-4827

JOURNAL OF APPLIED INFORMATION, COMMUNICATION AND TECHNOLOGY

Published by:
Research Center of Technology and Innovation
Directorate of Academic Research and Community Services
Swiss German University

Developing a Scalable and Accurate Job Recommendation System with Distributed Cluster System using Machine Learning Algorithm

Timothy Dicky, Alva Erwin, Heru Purnomo Ipung

Department of Information Technology, Swiss German University, Tangerang 15143, Indonesia

Article Information

Received: 29 January 2021
Accepted: 17 March 2021
Published: 18 March 2021
DOI: 10.33555/jaict.v7i2.108

Corresponding Author:

Name: Timothy Dicky
Email: timothy.dicky@student.sgu.ac.id

ISSN 2355-1771
eISSN 2723-4827

ABSTRACT

The purpose of this research is to develop a job recommender system based on the Hadoop MapReduce framework to achieve scalability of the system when it processes big data. Also, a machine learning algorithm is implemented inside the job recommender to produce an accurate job recommendation. The project begins by collecting sample data to build an accurate job recommender system with a centralized program architecture. Then a job recommender with a distributed system program architecture is implemented using Hadoop MapReduce which then deployed to a Hadoop cluster. After the implementation, both systems are tested using a large number of applicants and job data, with the time required for the program to compute the data is recorded to be analyzed. Based on the experiments, we conclude that the recommender produces the most accurate result when the cosine similarity measure is used inside the algorithm. Also, the centralized job recommender system is able to process the data faster compared to the distributed cluster job recommender system. But as the size of the data grows, the centralized system eventually will lack the capacity to process the data, while the distributed cluster job recommender is able to scale according to the size of the data.

Keywords: *Distributed Cluster System, Hadoop MapReduce, Machine Learning,*

1. Introduction

In 2030 – 2040, Indonesia is predicted to experience a Demographic Bonus period, where the number of population that is in working age (15-64 years old) is exceeding the number of population that is not in working age *(Bappenas, 2017)*. To exploit this demographic bonus period, the Indonesian government, as stated in RPJMN, will focus on two issues: current workforce and education. The government policies to improve the Indonesian workforce include the standardization of competencies data across sectors to support the open market. *(Indonesia, Republik, 2015)*

With such a big amount of workforce expected in Indonesia, an online job marketplace will be beneficial to many. Online job marketplace is a market that connects employers that are searching for employees and applicants that are looking for jobs. The job marketplace is not a new technology, in 2017, a popular employment website known as Linked In has reached 500 million members with 40% users are using Linked In daily *(Gallant, 2018)*. Currently, many online job marketplace is available in Indonesia and is used widely by Indonesian citizen such as jobsdb.com, jobstreet.com, karir.com and many more.

One of the main features of an online job marketplace is a job recommendation system. The job recommendation system recommends a potential job for job seekers. Job recommendation system can also give potential candidate recommendation to the job poster. This job recommendation system is usually based on a string search method or boolean search method to find a match between job vacancies and job seekers. Unfortunately, job marketplace that is using such algorithm tend to give a result with a low accuracy level since such a search method is not sufficient to match a person capability and the job requirement *(Al-Otaibi & Ykhlef, 2012)*.

Another big challenge of a job recommendation system is the amount of data involved. If a job poster requests a potential candidate recommendation, the recommender system will have to evaluate an enormous amount of potential candidates to provide the job poster an appropriate recommendation. Such evaluation process would require a costly computational power. As the data grows, a recommender based on centralized system architecture will lacks the capabilities to scale up well with the size of the data. Based on this research problem, this research has objective to prove that Hadoop MapReduce can scale linearly when it processes a big amount of data and thus able to increase the scalability of a job recommendation system that implements a machine learning algorithm. Another objective is to find out which machine learning algorithm that produces the most accurate job recommendation result.

2. Related Works

A research by Al-Otaibi stated that a job recommender system that implements boolean-search or string-search algorithm provides inaccurate result as it fails to consider various factor related to job fit of an applicant, they purpose that recommender system that implements similarity measure algorithm can better calculate the similarity between jobs and applicants thus increasing the accuracy of job recommender system. (Al-Otaibi & Ykhlef, 2012)

Apache Hadoop is a software designed for distributed system known for it's capability to scale linearly with the size of the data. A research by Fuad shown that the performance of MySQL to perform a query is faster compared to a Hadoop system (HIVE) for small data

size. But as the data size grow, Hadoop system will be able to scale better compared to MySQL as it's able to perform the query faster. (Fuad, Erwin, & Ipung, 2014)

3. Proposed Method

To develop an accurate and scalable job recommender, 2 experiments are performed. The first experiment used test data of applicant job preferences based on a set of applicant data and job post data to determine which similarity measure algorithm that provides the most accurate job recommendation result when it is implemented inside the machine learning algorithm that is used as the job recommender.

Then to test the scalability of the job recommender program, the second experiment used a large quantity of applicants data and job posts data as an input to the job recommender. The time consumed, and also the limit of the recommender program is observed in this experiment.

3.1 Applicants and jobs data

Test data used to measure the accuracy of the job recommendation algorithm is gathered via short interview with 4 respondents that is currently working in various IT fields. To do this, 20 IT job posts are gathered from an online job marketplace. The respondents are asked to choose 3 jobs out of 20 that is most suitable to them, which is recorded. Their job preferences are used as a test data to determine the accuracy of the similarity measure algorithm.

To test the scalability of the recommender system, a big applicants and jobs data is required. Since the data structure of applicants and jobs data follows the MySQL database structure used in this experiment, data set used for this purpose are generated via a JavaScript script to match the custom data structure needed. 30.000 job posts data are generated, along with their skill and experience requirement. And 8 million applicants data are generated, along with their skill and experience data, separated in chunk of files so that the job recommender program can take a partial amount of the applicant data.

3.2 Centralized job recommender

The centralized recommender is constructed using Java programming language and is run inside JVM environment. The program takes the applicant's skill data and experience data, job post's skill requirement data and experience requirement data as an input and produce a job recommendation data as an output. The centralized recommender is constructed with block programming model, the system architecture for the centralized job recommender is shown in figure 1.

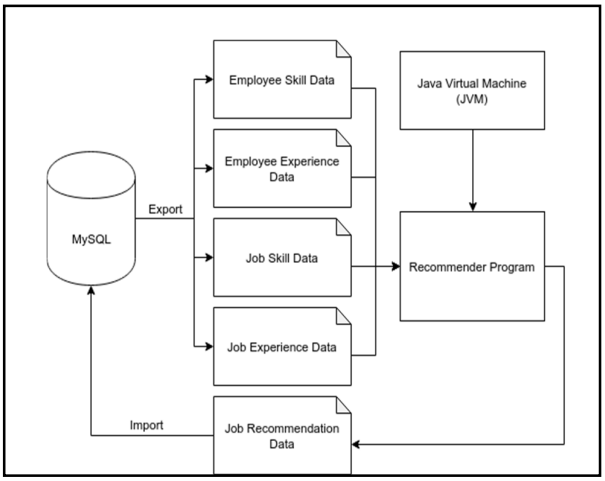

Figure 1. Centralized Job Recommender System Architecture

3.3 Distributed cluster job recommender

Hadoop MapReduce Java library will be used to construct the distributed cluster job recommender. There are 3 Hadoop environments set up used in this experiment. Pseudo-distributed, distributed cluster with 1 master node and 4 worker nodes (referred as 4 nodes cluster), and distributed cluster with 1 master node and 6 worker nodes (referred as 6 nodes cluster). The input data is extracted from MySQL database into HDFS, which then are read by the recommender program. The program architecture for the distributed cluster job recommender is shown in figure 2.

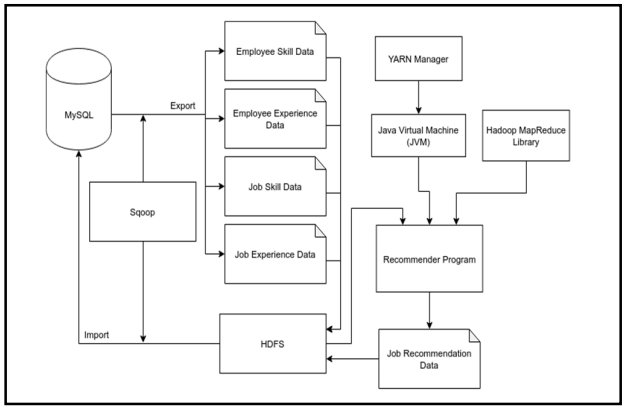

Figure 2. Distributed Cluster job Recommender System Architecture

4. Experimental Results

4.1 Accuracy result on similarity measure algorithm

The test data gathered from the interview process is used to measure the accuracy of similarity measure algorithm used in the machine learning algorithm. The job data and the applicant data are inputted to the recommender system to obtain the job recommendation produced by each similarity measure. The job recommendation result of each algorithm are listed in Table 1.

It is also worth mentioning that the test data obtained from respondent 1 has some inconsistencies in it. The skill set of respondent 1 is not correspond with his list of

experiences. Since the recommender is designed to match applicants and jobs based on their similarity of skills and experiences, it is to be expected that the job recommendation output quality for respondent 1 is poor, and respondent 1 data can be considered to be an outlier in this case.

Based on the output, cosine similarity measure provides the most accurate result compared to the other similarity measure.

Table 1. Job Recommendation of Each Similarity Measure Algorithm

	Expected Output		
Respondent 1	Job 5	Job 6	Job 15
Respondent 2	Job 1	Job 3	Job 8
Respondent 3	Job 1	Job 2	Job 4
Respondent 4	Job 11	Job 12	Job 17

	Cosine Similarity		7 of 12
Respondent 1	Job 15	Job 13	Job 14
Respondent 2	Job 4	Job 1	Job 3
Respondent 3	Job 4	Job 1	Job 13
Respondent 4	Job 11	Job 3	Job 12

	Jaccard Distance		6 of 12
Respondent 1	Job 13	Job 14	Job 15
Respondent 2	Job 1	Job 8	Job 13
Respondent 3	Job 1	Job 4	Job 13
Respondent 4	Job 3	Job 6	Job 11

	Euclidean Distance		6 of 12
Respondent 1	Job 1	Job 4	Job 6
Respondent 2	Job 1	Job 4	Job 8
Respondent 3	Job 1	Job 4	Job 6
Respondent 4	Job 3	Job 6	Job 11

4.2 Analysis on the centralized job recommender system

The experiment shows that the centralized recommender computation speed is almost linearly scaled to the amount of data used, for data amount up to 3,2 million applicants and 30.000 jobs. Afterward the performance drops slightly as the chart on Figure 3 shows that there are some skew on the chart after 3,2 million applicants.

Contrary to the initial assumption based on the conclusion of the paper published by Arsan (Arsan, Koksal, & Bozkus, 2016) the centralized recommender system does not appear to experience major performance issue expected. It was expected that the centralized recommender system will have a spike in the chart as the data amount grows.

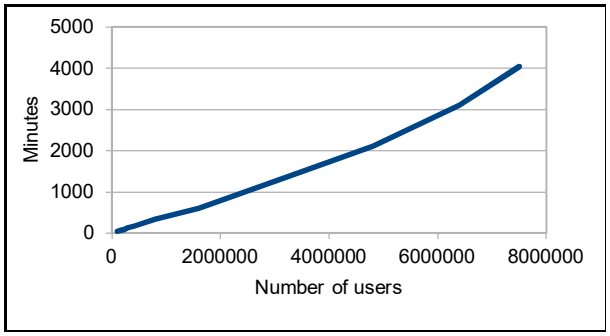

Figure 3. Processing Time for Centralized Recommender

However, there are some limitation discovered from the experiment. First as the data amount grow, the memory consumption of the program increases. Since the centralized recommender use a single machine to do the processing, there is a limit to how much data that a centralized system can handle. In this experiment, the maximum capacity allocated for the centralized recommender is 8196 MB (~8 GB). For that memory capacity, the maximum number of applicants the system can handle are around 7,5 million applicants. An increase to the amount of applicants will eventually led to the JVM throwing "OutOfMemory" error.

Second, the job recommendation output produced by the system get exponentially large as the data amount grow. In this case, the job recommendation data produced by the system take up around 459 MB of hard drive space (around 7.250.000 rows of job recommendation data) for each 100.000 applicants data evaluated against ts 30.000 job posts. Similar to the first problem, there are limits to how much hard disk space a single machine can handle.

In conclusion, although the performance of the centralized system is able to scale almost linearly to the amount of the data, the memory and the hard drive of the machine will eventually unable to handle the applicant data, which proves that the centralized system faced scalability issue.

4.3 Analysis on the distributed cluster job recommender system

Due to the limitation of the platform used in this experiment, processing time data gathered for the distributed cluster stops at 400.000 applicants data.

The pseudo-distributed job recommender system does not performs well and the processing time increased greatly compared to the fully distributed cluster system.

The distributed 4 nodes and 6 nodes both able to scale linearly to the amount of data. Furthermore, the 6 nodes cluster appear to have slightly less increment of processing time compared to the 4 nodes. And it is shown that the processing time can be reduced by adding more worker nodes in the Hadoop cluster.

Furthermore, by adding more worker nodes, the capacity of both memory and storage of the job recommender system can be increased based to the needs. This capability to add more memory and storage by simply adding more nodes to the cluster greatly increase the scalability capability of the distributed cluster job recommender.

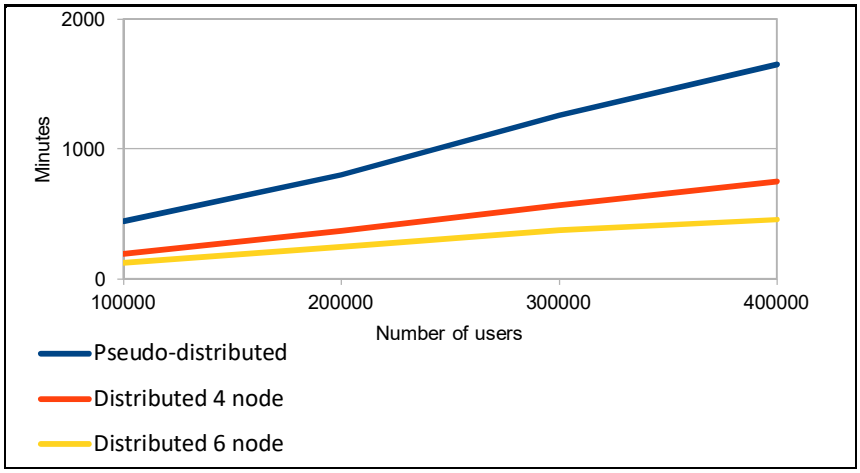

Figure 4. Processing Time for Distributed Cluster Recommender

5. Conclusions and Future Works

In the experiment, a job recommender system is built by comparing various similarity measure to determine the similarity measures algorithm that produces the most accurate result. Cosine similarity is proved to be the similarity measure that is best suited to be used to produce a job recommendation in comparison with euclidean distance and jaccard similarity.

Based on the experiments, it is shown that the centralized job recommender system is able to produce the job recommendation result faster compared to the distributed cluster job recommender system. However, one big problem of a job recommender system is the amount of data involved. As the amount of data increases, the centralized recommender system will begin to struggle to process the data because there is a limit to how much data a centralized recommender system can handle. By using Hadoop MapReduce, the job recommender system is able to scale up linearly for the computation of big data whereas the centralized system cannot. Also, HDFS can solve the problem with the size of the data by providing a scalable file storage system that is able to increase the capacity by adding more nodes.

Hadoop MapReduce can increase the scalability of a job recommender system. Aside from Hadoop MapReduce, there exist another distributed cluster system. For future works, the author propose that the performance of Hadoop MapReduce to be compared to its alternative, such as Apache Spark, Apache Storm, and H2O, to find out if there is a better alternative that could increase the scalability of a job recommendation even further compared to Hadoop MapReduce.

Furthermore, the time consumption needed for the job recommender system to process the data can still be improved by implementing better recommendation algorithm, which also needs further research.

References

Al-Otaibi, S., & Ykhlef, M. (2012). Job recommendation systems for enhancing e-recruitment process. *Proceedings of the International Conference on Information and Knowledge Engineering*, pp.1-7. Nevada.

Arsan, T., Koksal, E., & Bozkus, Z. (2016). Comparison of collaborative filtering algorithms with various similarity measures for movie recommendation. *International Journal of Computer Science, Engineering and Applications,* 6(3), pp.1-20.

Bappenas. (2017). *Siaran Pers Bonus Demografi 2030-2040: Strategi Indonesia Terkait Ketenagakerjaan dan Pendidikan.* Jakarta: Bappenas.

Fuad, A., Erwin, A., & Ipung, H. P. (2014). Processing performance on Apache Pig , Apache Hive and MySQL cluster. *Proceedings of International Conference on Information, Communication Technology and System (ICTS)*, pp.297–302.

Gallant, J. (2018). *45 Eye-Opening LinkedIn Statistics for B2B Marketers in 2018*. Retrieved from Foundation Inc, 2018: https://foundationinc.co/lab/b2b-marketing-linkedin-stats/

Indonesia, Republik. (2015). *Rencana Pembangunan Jangka Menengah Nasional 2015-2019.*