International Journal of Contemporary Computer Research (IJCCR), Vol.2 Issue.1 (June, 2024)

ENHANCED DYNAMIC INFRASTRUCTURE AS A SERVICE MODEL (EDIAAS) FOR SELECTING SERVICE PROVIDER

Yazeed Al Moaiad[1], Zainab Abu Bakar[2]
Al-Madinah International University, Malaysia
[1] yazeed.alsayed@mediu.edu.my
[2] zainab.abubakar@mediu.edu.my

Abstract—**The number of cloud service providers expanding their services differs from one another. These have caused difficulty for users to choose the best services for a particular application. It is a tedious process for a user to search and select service providers before a chosen service can be extracted and compiled according to user preference. This process will repeat until all the desired services by the user are compiled. Then the user has to rank the providers and make the decision about which provider fulfills his need. Unfortunately, the user has to search for the information again as it will be updated by the service provider. In this paper, we propose an Enhanced Dynamic Infrastructure as a Service Model (EDIAAS) for selecting service providers based on user needs that incorporate important techniques such as worker role, cache redis, and SignalR. The prominent services considered will be infrastructure as a service focusing on the speed of the central processing unit (CPU), the size of random-access memory (RAM), the size of solid-state drive (SSD), the bandwidth in bits per second (bit/s), and the cost of service. The experiments conducted show that incorporating techniques such as worker role, cache redis, and SignalR has increased system efficiency and reduced the time to display up-to-date information and the results instantly. This will ease the efficiency of the user's search and ranking task in selecting cloud service providers.**

Keywords- Cloud Service Providers, User Preference, Infrastructure As A Service, Efficient Search

I. INTRODUCTION

[1] has proposed a general structure of Cloud service publication, discovery, and selection however there is no methodology proposed in their work. The Cloud service selection can usually be solved by Cloud service comparison because of target execution investigation [2]. CloudCmp [3], an efficient comparator, can be connected to look at three parts of the execution and expense of a Cloud, that is, flexible registering, diligent capacity, and intra-Cloud and wide range organizing [4]. These examinations are acknowledged by an arrangement of standard benchmark instruments, whose outcomes show the target evaluation of a Cloud.

[5] consider the differences in Cloud the measurements of versatility, cost, top load, and adaptation to internal failure. Another discourse of Cloud benchmark testing is introduced by [6]. In their work, they call attention to that the execution markers given by Cloud suppliers may not be sufficient to judge the genuine execution of a virtual machine and propose another execution estimation system, which considers the sorts of services executed on a virtual machine for Infrastructure-as-a-Service mists. As of late, some outsider associations such as CloudHarmony have begun to offer Cloud observing and benchmarking services [7]. In customary e-trade or e-service situations, service choice for the most part relies on the notoriety-based trust assessment of services. Contrasting with right-on-time trust assessment approaches because of processing solitary trust esteem for a service [8] proposed a few trusts vector-based assessment approaches, where a trust vector is ascertained to reflect both the present dependability of a service and its trust trend. Such trust values or vectors are all evaluated from evaluations that speak to the subjective appraisal of services given by service buyers, keeping in mind the end goal to consider subjective parts of a Cloud service [9].

[10] proposed a basic system for checking Cloud execution because of client criticism, in which the execution of a Cloud service is observed and anticipated by clients' input. Their methodology just considers Cloud clients' subjective evaluation. There is no component to check the dependability of clients' criticism. What's more, the target appraisal of a Cloud service is not considered in their structure. Another answer for the Cloud service choice issue is to demonstrate the issue as a multi-criteria choice-making (MCDM) issue [11], which can be regularly fathomed by Analytic Hierarchy Process (AHP) [12].

[13] concentrated on the choice of Software-as-a-Service mists in light of AHP. Five elements, that is, usefulness, construction modeling, ease of use, seller notoriety, and expense, are considered in their methodology. It ought to be noticed that every one of these elements aside from expense can barely be quantified by a goal measure. Hence, their methodology is still for the most part because of subjective appraisal. Another AHP-based Cloud examination methodology is proposed by [13], In their work, they endeavor to institutionalize the execution qualities for Cloud correlation. Be that as it may, the institutionalization for a few qualities, for example, supportability and straightforwardness, are

excessively basic, making it impossible to reflect the intricate circumstances of Cloud services in this present real world.

The services provided are subject to changes and thus the user cannot rely on the decision made earlier. Thus, the first objective of this research is to construct a Dynamic Infrastructure, as a Service Model (DIAAS) for selecting service providers based on up-to-date user preferences. Secondly, to improve time efficiency, the DIAAS model needs to be incorporated important techniques such as worker role, cache redis, and SignalR. The new model is known as the Enhanced Dynamic Infrastructure as a Service Model (EDIAAS). Finally to show the efficiency of EDIAAS [14, 15].

II. CONSTRUCTION OF DYNAMIC IAAS MODEL (DIASS)

The steps in the construction of DIAAS are shown in Figure 1. The data values for service providers are dynamically retrieved and grabbed using web services. This is automatically carried out by an intelligent tool, ITOOL since there is no standard tool available. ITOOL will visit the data page on each site, and do intelligent searching and recording for the highest value plan. This will ensure to capture any changes in the arbitrary data and process the data to the standard data. The tool will continue looping to examine the values of each row on each table, inside each provider site [14, 15]. The intelligence tool algorithm will perform the following steps;

1. Fetching the pricing web page and detecting all the tables on that page.
2. Detecting the needed pricing tables using regular expressions to find the needed data.
3. 3Looping over all of the tables and data sets until reaching the table with the highest values in CPU, RAM, SSD, bandwidth, and cost.
4. Storing the highest value inside a new variable and populating using standard way JSON.
5. Processing all the HTML tags using special libraries. Importing the data in text format, and keeping track of the location of the table and cell.

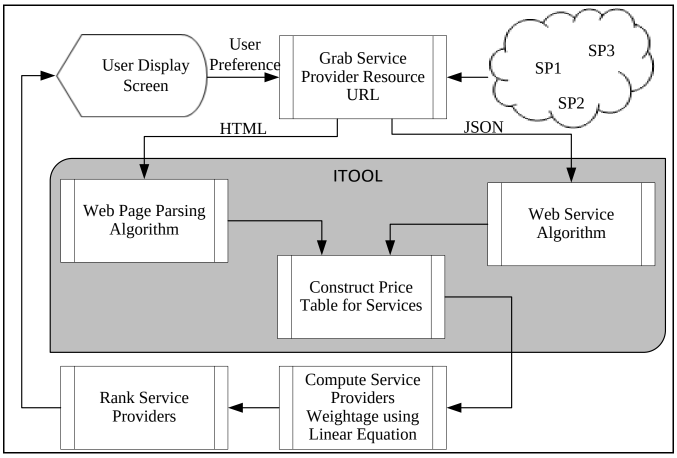

Figure 1. Dynamic Infrastructure as a Service Model (DIAAS).

III. CONSTRUCTION OF ENHANCED DYNAMIC IAAS MODEL (EDIASS)

As mentioned earlier, DIASS searches and ranks providers according to user providers dynamically. This model can be further improved by reducing time using important techniques such as worker role, cache redis, and SignalR. Figure 2 shows the phases and steps in the construction of EDIAAS. There are four phases in implementing EDIAAS. Note the usage of ITOOL constructed in DIAAS is employed in EDIASS [14, 15].

A. Phase 1: Connect to the server

1. Connect to the cloud provider and retrieve data during server startup.
2. Store the retrieved data in the cache redis.
3. Retrieve the data from the cache Redis.
4. Display the retrieved data to the user.
5. Trigger the worker role to perform the following tasks in the background in parallel.

B. Phase 2: If web service

1. Invoke the cloud provider web service.
2. Compare the retrieved data to the data stored in the cache redis.
3. If there is a mismatch;
   a. Update the cache Redis
   b. Notify the user of the updated data using SignalR technology

C. Phase 3: If web page parsing

1. Connect to the cloud provider and retrieve the pricing HTML page.
2. Store the retrieved pricing html page in memory.

3. Traverse the pricing HTML page.
4. Find the location of the required values based on the location of the table containing those values.
5. Compare the values found in the specified location to the values in the cache redis.
6. If there is a mismatch;
   a. Update the cache Redis.
   b. Notify the user of the updated data using SignalR technology.

Phase 4: Selecting

1. The user selects data preference.
2. The user submits the data to the server.
3. Use a linear equation algorithm to calculate the cloud provider's rankings.
4. Display the cloud provider's rankings to the user.

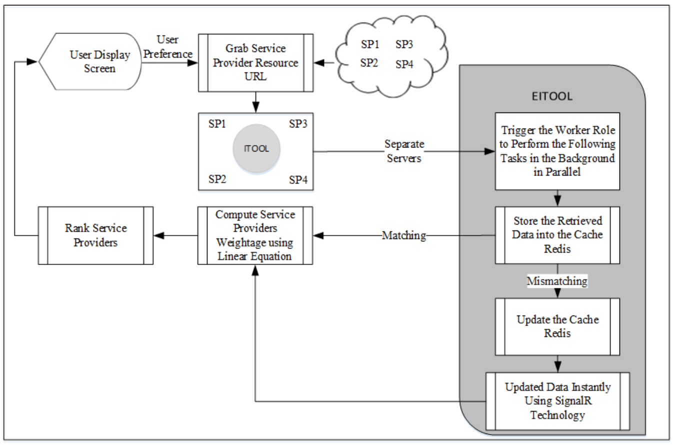

Figure 2. Enhanced Dynamic Infrastructure as a Service Model (EDIAAS).

IV. Results and Discussion

The two dynamic models, DIAAS and EDIAAS are compared in terms of three factors, that is, capacity, load, and time for a user whose that's score is values of 1-3. A score of 1 indicates the user requires a high level of the service; 2 implies a medium level and 3 indicates a low level. Table 1 shows an example of user requirements.

TABLE 1. REQUIREMENT LEVEL OF SERVICE NEEDS OF EXPERIMENTAL USER.

	SSD	RAM	CPU	Bandwidth	Cost/Month
User	3	2	2	2	3

Table 2 shows the performance measures of DIAAS and EDIAAS in terms of capacity, load, and time based on the experimental user's requirement. Capacity, load, and time are measured using the JMeter tool. Table 3 summarizes the differences between static and dynamic models.

TABLE 2. PERFORMANCE MEASURES BETWEEN DIAAS AND EDIAAS.

Factors	DIAAS Model	EDIAAS Model
Capacity	68%	93%
Load	500/responds	500/ responds
Time/Average	23.424/s	11.834/s

TABLE 3. PERFORMANCE BETWEEN DIAAS AND EDIAAS.

Factors	DIAAS Model	EDIAAS Model
Capacity	Acceptable	Capability
Load	Reasonable	Sustainable
Time	Fast	Instantly

The capacity designed is to handle the number of users and processes. DIAAS has a capacity of 68%, which is acceptable. In contrast, the EDIAAS model has a capacity of 93%. The load testing for 500 concurrent users is considered reasonable as the number approaching 450 loadings becomes slower and the errors increased to 32% when the users reach 490. Thus, DIAAS is considered a failure. While EDIAAS can load, continuously with only 7% error. Finally, the response time is the average time to fetch the homepage is 23.424/s for DIAAS and 11.834/s for EDIAAS. This implies that the user instantly receives the result although the number of users is less than 500.

V. Service Conclusion And Future Works

The experiments conducted show that incorporating techniques such as worker role, cache redis, and SignalR has increased system efficiency and reduced the time to display up-to-date information and the results instantly. This will ease the efficiency of the user's search and ranking task in selecting the cloud service provider. The model not only helps users to select the best service provider but also to understand their requirements and serves to assign a ranking to provider companies to make the process of researching for providers easier.

Future works will be more comprehensive and take into account the reputation of providers, which contributes to a high trust efficiency. It is intended to apply this in the model in a practical way. It also helps providers companies to judge their performance by trying to face the challenges that come with increased users with various priorities of services.

References

[1] Binnig, C., Kossmann, D., Kraska, T., &Loesing, S. (2009). How is the weather tomorrow?: to-wards a benchmark for the Cloud. In

Proceedings of the Second International Workshop on Testing Database Systems (p. 9).

[2] Garg, S. K., Versteeg, S., & Buyya, R. (2013). A framework for ranking of cloud computing services. Future Generation Computer Systems, 29(4), 1012-1023.

[3] Godse, M., & Mulik, S. (2009, September). An approach for selecting software-as-a-service (SaaS) products. In Cloud Computing, 2009. CLOUD'09. IEEE International Conference on (pp. 155-158). IEEE.

[4] Goscinski, A., & Brock, M. (2010). Toward dynamic and attribute-based publication, discovery, and selection for cloud computing. Future generation computer systems, 26(7), 947-970.

[5] Jain, H. K., Tanniru, M. R., & Fazlollahi, B. (1991). MCDM approach for generating and evaluating alternatives in requirement analysis. Information Systems Research, 2(3), 223-239.

[6] Leitner, P., & Cito, J. (2016). Patterns in the Chaos—A study of performance variation and predictability in public IaaS clouds. ACM Transactions on Internet Technology (TOIT), 16(3), 15.

[7] Lenk, A., Menzel, M., Lipsky, J., Tai, S., & Offermann, P. (2011, July). What are you paying for? performance benchmarking for infrastructure-as-a-service offerings. In Cloud Computing (CLOUD), 2011 IEEE International Conference on (pp. 484-491). IEEE.

[8] Li, L., & Wang, Y. (2008, September). A trust vector approach to service-oriented applications. In Web Services, 2008. ICWS'08. IEEE International Conference on (pp. 270-277). IEEE.

[9] Li, L., & Wang, Y. (2010). Subjective trust inference in composite services. In AAAI (pp. 1377–1384).

[10] Li, A., Yang, X., Kandula, S., & Zhang, M. (2011). Comparing public-cloud providers. IEEE Internet Computing, 15(2), 50-53.

[11] Li, A., Yang, X., Kandula, S., & Zhang, M. (2010, November). CloudCmp: comparing public cloud providers. In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement (pp. 1-14). ACM.

[12] Li, L., Wang, Y., & Varadharajan, V. (2009, July). Fuzzy regression-based trust prediction in ser-vice-oriented applications. In International Conference on Autonomic and Trusted Computing (pp. 221-235). Springer Berlin Heidelberg.

[13] Rehman, Z., Hussain, O. K., Parvin, S., & Hussain, F. K. (2012, July). A framework for user feed-back based cloud service monitoring. In Complex, Intelligent and Software Intensive Systems (CISIS), 2012 Sixth International Conference on (pp. 257-262). IEEE.

[14] Al Moaiad, Y., Bakar, Z. A., & Al-Sammarraie, N. A. (2016, October). Prioritization tool of Cloud Computing service provider based on user requirement. In 2016 IEEE Conference on Open Systems (ICOS) (pp. 36-41). IEEE.

[15] Moaiad, Y. A., Bakar, Z. A., & Al-Sammarraie, N. A. (2018). Constructing dynamic infrastructure as a service model (diaas) according to user preferences. In Regional Conference on Science, Technology and Social Sciences (RCSTSS 2016) Theoretical and Applied Sciences (pp. 185-194). Springer Singapore.