International Journal of Research in Engineering
www.engineeringpaper.net
Online ISSN: 2664-8784; Print ISSN: 2664-8776; Impact Factor: RJIF 5.28
Received: 10-06-2021, Accepted: 15-06-2021, Published: 05-07-2021
Volume 3; Issue 1; 2021; Page No. 24-30

Grey wolf optimizer with learning algorithm for web of services

[1]Haneen and [1]Shravan Kumar
[1]Ryan College of Engineering, Madurai, Tamil Nadu, India

DOI: https://doi.org/10.33545/26648776.2021.v3.i1a.60

Abstract

Using a web service is like switching to a new operating system for the Internet. An essential component of the web service concept is a brokerage system that facilitates the publishing and retrieval of services from a searchable repository. In this study, we use a Distributed Constraint Optimization Problem (DCOP)-based agent-based approach to do this. Finding the best service based on device-specific quality-of-service criteria is the main objective of this research. We use a DFS (Depth First Search) tree-based parallel search technique. Using criteria including availability, affordability, safety, reliability, reaction time, and power consumption, DCOP allows several agents to work together to choose the best service to employ. As a result of using the DCOP approach, the devices' collected global constrain characteristics are validated. The fact that the DCOP algorithm generates a linearly proportional amount of messages is its primary strength. Here, messages of type UTIL and VALUE are exchanged. Agents may converse with their parents using UTIL messages and with their children using VALUE messages. The next step is to choose the best online service by using the agents. A multi-agent global constraint system, it expedites the acquisition of online services.

Keywords: Web service, web service discovery, GWO, QOS and agents

Introduction

A web application is a protocol that allows programmes to interact with an application programming interface. When many systems need to connect with one another through a network, a web service may assist by providing an API access to the business logic. The entity offering the service is known as a web service provider, while the entity receiving the service is known as a web service consumer. Learn some key terms associated with online services with the aid of this brief guide. Eclipse-based web service development lessons will become available soon [1]. A web service is a way for software to share its resources with other programmes via the Internet by using common Internet protocols. Web services are built on standardised interfaces that allow them to interact across platforms, languages, and operating systems. This makes them a great method for developing networked applications that must support several kinds of hardware. This section describes the common tools for creating web services, as well as the additional features made possible by asynchronous web services [2].

Standard Technologies

Online services conform to established norms, allowing them to make their resources easily discoverable. Online service:

- Publicly describes its own functionality through a WSDL file
- Communicates with other applications via XML messages, often formatted with SOAP
- Employs a standard network protocol such as HTTP

In fields like networking and distributed systems, Quality of Service (QoS) is an overarching measure for summarizing the features of such systems. We provide a lightweight method for dealing with the Web Service QoS issue by making use of the established Web Service Distributed Management (WSDM) & Web Service Ping. The Quality of Service (QoS) attributes of a Web Service may be made public using the MOWS component of WSDM. Part two of WSDM, Quality of Service (QoS) monitoring for IT resources, is implemented through Management via Web Services (MUWS). Quality of service in the context of the server, the application server, the network, etc. [3]. Web Service Ping is a straightforward method for diagnosing cross-departmental Web service latency and availability. We suggest standardizing the Ping procedure for all Web Services as a result.

Web Service selection may make advantage of the quality-of-service information gathered by MOWS, MUWS, and Web Service Ping [4]. Delegation Web Service as picker is a new selection architecture we presented. The Delegated Web Service as selector is an improved alternative to Web Service Broker, consumer, and QoS enhanced UDDI for creating Web Service load balancing, and it may also improve Web Service security [5].

Current methods for selecting a provider based on QoS presuppose, without fail, that this information will be accurate and reliable. QoS qualities offered by service providers, on the other hand, might have implausibly high values in certain cases, as providers may exaggerate the quality of their services to entice more customers. In order to improve performance, it is possible to raise the maximum response of these services while keeping the invocation rate below a predefined limit [6]. As a result, it's important to be able to objectively and effectively evaluate a service provider's reputation so that consumers may use it as a reference when selecting. This research presents a trust Web service repute assessment framework based on the similarity between the true values and the stated values to guarantee the objectiveness and impartiality of a Web application reputation evaluation [7]. To begin, an agnostic QoS assessment tool for Web services was created, one that doesn't need manual intervention from either service providers or end users. Second, the Web service's reputation is evaluated by comparing the advertised and actual quality of service (QoS) values and computing the resulting similarity. Finally, a series of tests is shown, demonstrating that this method may accurately assess the service provider's reputation, hence improving the efficiency of the services selection process [8-10].

Literature Review

The use of Web services delivered through the internet as part of a collaborative production system known as "cloud manufacturing" is gaining ground across the world. It will reveal the immense value hidden in the mountains of data being produced by manufacturers. One fundamental aspect of cloud manufacturing is the challenge of quality-of-service-aware Web service composition (QWSC), or choosing the right service for the right part of a service composition from a set of functionally equivalent services so as to meet the users' end-to-end QoS restrictions. In order to provide customers with more options, we present a unique QWSC approach based on multi-objective optimization. First, QWSC is framed as a multi-objective optimization problem, with each optimization goal focusing on either QoS effectiveness or QoS risk (Variation compared to the user's QoS demand). To address this scenario, we next design and implement an effective -dominance multi-objective evolutionary algorithm (EDMOEA). At last, experimental results validate the efficacy and efficiency of the suggested approach for the widespread QWSC issue. Value-added services may be developed with the help of Web service composition, a crucial technology for integrating existing services. Mass services on the Internet that have the same features but varying QoS values are becoming more common as a result of the fast growth of Service Computation, Cloud Services, Big Data, as well as the Internet of Things. In addition, the dynamic nature of a service's quality of service (QoS) is a result of the unpredictability of its application environment, making it difficult to compose dynamic Web services that can be relied upon. In order to solve this problem, this article provides a two-step method for composing dynamic Web services reliably [11]. After collecting each service's historical QoS values, the suggested method, Cultural Genetic Algorithm, chooses the best K composition schemes for those services (CGA). Then, the top K schemes' component services are selected to be used as dynamic service composition candidates. This process may narrow down the pool of potential services for use in service composition on the fly. Converting the global optimum issue of service selection into the an optimal solution problem of service discovery using the CGA algorithm enhances the flexibility of dynamic service construction and provides the option to predict QoS values of services prior to service selection. Second, while the load balance process is underway, choose the best service for each task, the QoS levels of every candidate service are forecasted using the enhanced case-based reasoning. QoS prediction is a powerful tool for making the composites Web service more reliable. The experimental evidence demonstrates the practicality and efficiency of the suggested approach. Quality of service (QoS), which basically characterizes non-functional qualities of web services, is often used in service development since online services have become a popular way for building software over the Internet. Composing Quality of Service (QoS) from huge candidate sets is a difficult multi-objective optimization issue since QoS is a composite of ideas. To narrow the possible solutions for a given service assembly, we use a Pareto-optimal perspective in this research. Our concept's viability is shown by introducing a Pareto solid modeling for QoS-aware service design and then theoretically investigating its link with the commonly used utility function model [12]. Several QoS attribute aggregation methods in service composition are explored systematically, & QoS-based dominance relationships are created for both candidates and processes. A candidate-based partial selection method for service composition is described, which takes use of candidate-level constraint validations and candidate-level dominance connections to reduce the number of potential solutions. In addition, a parallel strategy is developed, which may drastically cut down on the search space and provide impressive performance improvements. The effectiveness of our method is confirmed by both theoretical analysis and empirical testing on simulated and real-world data, respectively. The focus of this article is on the enabling technologies, protocols, & application concerns that are central to the IoT. New RFID and smart sensor technologies, as well as advances in communication and Internet protocols, make the IoT possible. The primary idea is to create a new category of applications by the direct cooperation of smart sensors with little human intervention. The present revolution in online, mobile, and M2M technologies might be seen as the IoT's incipient stage. It is anticipated that in the next years, the IoT will link physical items together to facilitate intelligent decision making, bridging disparate technologies in the process. At first, this presentation gives a broad introduction to the Internet of Things. We then provide a high-level review of certain technical aspects of the technology, protocols, and applications that make the Internet of Things possible. To help researchers and application developers quickly understand how the various protocols work together to provide the desired functionalities, we aim to provide a more 3 outlines of all the relevant protocols and application issues than previous survey papers in the field [13]. This will save them time and prevent them from having to dig through RFCs as well as the standards specifications. We also summarise relevant studies and offer a summary of a few of the most pressing issues facing the Internet of Things as stated in the current literature. We also investigate how the IoT connects to other cutting-edge fields, such as big data analytics & cloud/fog computing. In addition, we call attention to the need of fostering deeper horizontal integration amongst IoT services. Finally, we illustrate that how aforementioned protocols combine to provide the promised IoT services by providing specific service usage cases. Common in the field of ubiquitous computing, proposed framework is the process of constructing more advanced services from simpler ones (both in terms of hardware and software). To provide customers the greatest service possible, service composition should think about their quality of service expectations. Careful service selection is required when composing services to meet customers' desired quality of service (QoS). Service discovery with global QoS restrictions is a computationally intensive problem (i.e., user-imposed criteria for the whole job). This challenge is compounded in ubiquitous environments by their inherent uncertainty, sparse computer resources, and stringent time constraints. To address this issue, we provide QASSA, a robust decision-making algorithm that provides the foundation for QoS-aware service discovery in ubiquitous environments. The effects of QASSA are far-reaching. By initially recasting service selection under global Qos requirements as a collecting optimization task, it makes advantage of recent developments in the area of multi-objective optimization. Second, QASSA employs clustering techniques, namely the K-Means method, to solve this problem. Lastly, QASSA is designed in two flavours, a centralized flavour for use with centralized infrastructure and a distributed flavour for use in distributed, ubiquitous situations. Results from experimental studies demonstrate QASSA's efficacy and efficiency. The proliferation of potential applications for IoT technology has contributed to the field's rapid expansion in recent years. IoT infrastructure must adhere to a broad variety of requirements due to the many applications it supports. Due to the diverse nature of the disciplines, the criteria required more complex systems with ever-shifting performance requirements. As a result, there is a plethora of IoT architectures, each featuring a distinct set of components, features, and jargon. This resulted in incompatibilities between different systems, which delayed development in the area as a whole. Reference architectures are a good tool for tackling these issues because they promote mutual understanding by providing a framework that may support different applications and, eventually, enable the recycling of work done in unrelated fields. This article presents a synopsis of European studys aimed at establishing such a common framework, discussing both their current status and their potential moving forward. With the proliferation of cloud-based offerings, consumers have raised fresh concerns about how to most effectively meet the dynamic choice problem for services. The purpose of this research is to develop a model for evaluating service operations that takes into account numerous requests and their interdependencies [14]. Functional quality, non-functional quality, & transactional quality are the three dimensions over which this model evaluates the service delivery process. Using a hybrid swarm - based optimization method that integrates the crossover and mutation operators of an evolutionary algorithms with a newly devised particle coding scheme, we have developed a technique for efficiently addressing the service selection problem. The experimental data we provide validates the viability and effectiveness of our proposed approach. One approach to creating distributed applications is known as "service-oriented computing" (SOC). Application interoperability development and research has been particularly active in the area of Web services composition throughout the last decade. Even though numerous books and articles have been written on the topic of building Web services, there are still many unanswered problems that need to be solved as new laptop paradigms such as Cloud computing, communal computing, and the Internet of Things arise. This article presents a summary of the most relevant standards, research prototypes, and platforms that contribute to the construction of Web services throughout all phases of their existence. A set of criteria for assessing such benchmarks, research prototypes, and platforms is outlined in the article. The paper also outlines several opportunities and challenges for further research into Web services architecture. As cloud computing is becoming more popular among those who use network services, web service providers have an incentive to develop and provide a wide range of services with different nonfunctional & functional (Quality of service) qualities. As the variety of services offered is only expected to grow, cloud service brokers will need to provide severe competition to attract and retain consumers. As a consequence of this competition, providing easy service selection & compilation while supplying composite cloud services becomes an NP-hard problem. Some of the most pressing issues to be addressed include how to choose the best services from the service pool, how to work around composition constraints, how to prioritise different aspects of service quality, how to account for the dynamic nature of the problem, and how to handle sudden changes in the services' and networks' characteristics. In this work, we undertake a systematic literature review to identify and raise important questions about the state of the art in this area of study. By classifying the research into four main groups based on the problem-solving approaches, defining the investigated quality of service standards, targeted objectives, and developing environments, we have gathered important data and statistics that may aid future study. Cloud computing has revolutionised the way in which web-based application services are hosted and distributed. The plethora of new cloud solutions and features by both large (e.g., Amazon Web Services and Microsoft Azure) and small businesses might leave authorities (e.g., applications designers as well as chief knowledge officers) feeling bewildered (e.g., Rackspace and Ninefold). The decision-making process is already complicated by the wide variety of service configurations and the constraints on quality of service (QoS) in application delivery. The earlier work we did to tackle this challenging issue included developing a semi-automated, extendable, ontology-based approach to identifying and choosing infrastructure services based on design-time constraints (e.g., the renting cost, the data centre location, the service feature, etc.). Our technique is expanded upon in this work by including end-to-end message latency and throughput as indicators of actual (run-time) QoS. Supporting the next generation of interactive online games, massive sensor analysis, and real-time mobile apps in the cloud requires optimization of service-level agreements based on such genuine QoS restrictions. By expanding on the well-known analytic hierarchy process methodology, we provide an inter judgement approach that accounts for the quality of services in real time. In order to make an informed decision when selecting an IaaS cloud provider, Quality of Service (QoS) limitations and needs may be specified by the user at both design time & runtime [15]. Our databases are then queried to determine which combinations of IaaS-level cloud computing have the highest likelihood of satisfying these criteria. We did a lot of testing to show that our method works.

Proposed Model

Problem Formulation

We provide a formal definition of the service-option problem in this analysis. It reframes two standard recommender approaches for service selection and presents a new advisor framework in which agents work together to evaluate service providers. Agents use this strategy to rate one another and weigh the significance of their own recommendations. The foundational approach to building an agent's reasoning inside a theoretical lattice, which enables the identification of suitable agents. We modify the popular brand assessments dataset Movie-Lens for use in testing service choices since there are presently no large-scale datasets available for testing. This aids in the assessment of several different approaches. The proposed approach limits communication, yet it beats state-of-the-art approaches on many standardised metrics of accuracy. The functionality of several web services is similar. Quality of service (QoS) and other non-functional characteristics should be used to help service consumers locate the online services they need. Web Service applications have an urgent need to figure out how to swiftly, accurately, and efficiently get services from huge and expanding service repositories. QoS-based web application discovery is vital to service-oriented architecture (SOA) since most applications like using services that are tailored to their specific requirements. We propose an interactive online service discovery adviser framework to make it simpler for the applicant to provide QoS requirements alongside functional demands. We provide a novel approach to developing a Quality of Service (QoS) based matching, rating, and selection algorithm for evaluating online services within an agent-based architecture. This article gives a recommended practise for identifying a suitable web service for a given set of requirements.

System Model
To choose the top services according to QoS standards, we suggested a DCOP (Distributed constraint optimization problem) method. First, we consider a network with a sizable number of IoT devices; this network communicates its quality of service requirements to the agent. The agent looks for the optimal services depending on the QoS qualities the device needs, such as cost, safety, dependability, speed of response, power consumption, and availability. Depth First Search (DFS) trees order is used by the agent to look for services; local communication is disabled. It uses two different sorts of messages-UTIL and VALUE-to communicate; if the device's quality of service standards are met, the web service is chosen.

Advantages
- We select best service based on the QoS criteria.
- Less execution time and faster process.

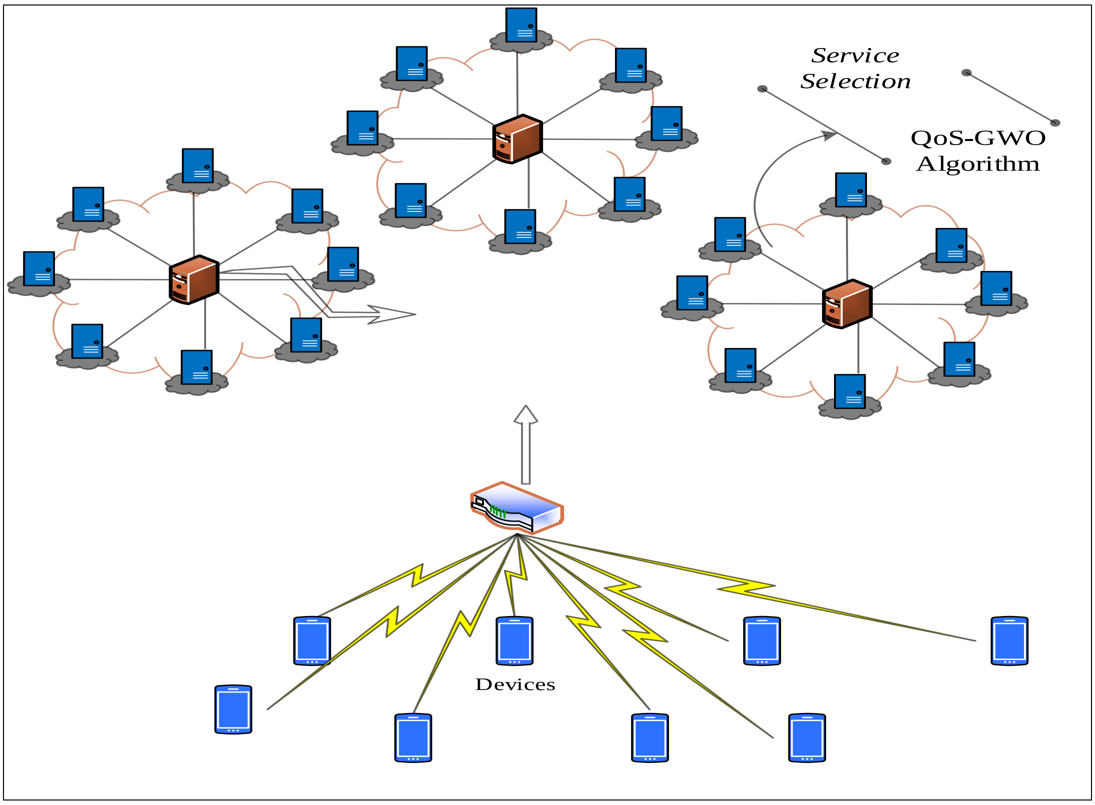

Fig 1: WoS Selection

QoS Modeling
In this approach, we choose a web service depending on a Quality of Service (QoS) parameter, such as the following two categories: Parameters, Both Static and Dynamic.

Static Parameter: Factors including response time, reliability, cost, and security all factor in.
- **Response time:** How long it takes for the service to process and react to the request.
- **Reliability:** The ability of the service to guarantee timely message delivery.
- **Price:** The cost associated with each service request.
- **Security:** Specifies the degree of protection provided by each service.

Energy level & availability are two of the parameters that make up this dynamic parameter.
- **Energy level:** Battery energy level of the object hosting service.
- **Availability:** It represents the accessibility of the service.

DCOP Optimization Algorithm
The DCOP Algorithm is being used in this study to solve a distributed constraint optimization issue. The global constrain characteristics provided by the IoT devices are met. Here, we employ a Depth First Search (DFS) trees to relay information to other agents and convey the necessary service request. The DFS operated in parallel to determine which service was ideal for each individual device.

Optimal Web service selection
In this work, we use the DCOP algorithms to prioritize services according to their quality of service characteristics. With a DFS architecture, agents and devices are only allowed to communicate with one another on a local level. It communicates in two distinct ways, through UTIL communications and VALUE messages.
- **UTIL Messages:** Bottom-up communication is the process through which agents relay information to their parent. Messages begin at the leaf nodes and spread outward along the branches of the DFS tree.
- **VALUE Messages:** The agents' message to their offspring is a kind of top-down communication. Based on the VALUE signal received from its parent, Agent calculates its ideal value.

A Grey Wolf Optimizer is a very recent swarm intelligence technique (GWO). The GWO took its cues for its hunting strategy from grey wolves. The GWA has been proved to be of good quality via early verification on benchmarking & real-world optimization difficulties. An optimization metaheuristic was inspired by the wolves' innate tendency to divide into four groups based on dominance. There are four official names for the different categories: alpha, beta, delta, and omega. The

alphas are the top dogs in the pack. The wolf just at bottom of a pack hierarchy are the ones who set the rules for the pack leaders up above. In the implementation of the optimisation method, the solutions with the greatest values of the fitness function is the leader; the solutions with the lowest values of the fitness function is the solutions, and so forth. Similarly, the symbol represents the population's third-best response. The surviving population of the planet fits this description. The original algorithm used a three-stage search process that is reminiscent of mining's exploration and extraction phases; the best possible answer is the wolves' dinner. Exploration occurred throughout the phases of following prey, refined exploration occurred during the hunting phase, and exploitation occurred during the striking phase.
In the commencement of a chase, which may be summarised as such, the wolf pack will attempt to encircle the victim:

$$\vec{D} = \left|\vec{C} \times \vec{X}_p(t) - \vec{X}(t)\right|$$
$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \times \vec{D}$$

where we've highlighted the current iteration with $t, \vec{A}$ is a random vector whose size is determined by the current iteration, $\vec{C}$ is a vector of random coefficients between zero and two. The $\vec{X}_p$ represents p as a population solution.
Variables $\vec{A}$ and $\vec{C}$ are defined by the following two equations:

$$\vec{A} = 2\vec{a} \times \vec{r}_1 - \vec{a}$$
$$\vec{C} = 2 \times \vec{r}_2$$

where $\vec{a}$ progressively decreases with time, beginning at 2 and ending at 0. Variables r_1, r_2 Symbolize a set of random vectors.
Hunting phase: Here we designate the top three options in descending order of fitness value as $\alpha, \beta,$ and δ. These options represent sections of the search universe that are very promising right now and need further investigation. The remaining options available to the populace X_i where $i = 1,2, \ldots, N-3$ and N is the population size, the following equations drive the population towards these three solutions:

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_1} \times \overrightarrow{X_\alpha} - \vec{X}\right|$$
$$\overrightarrow{D_\beta} = \left|\overrightarrow{C_2} \times \overrightarrow{X_\beta} - \vec{X}\right|$$
$$\overrightarrow{D_\delta} = \left|\overrightarrow{C_3} \times \overrightarrow{X_\delta} - \vec{X}\right|$$
$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1} \times \overrightarrow{D_\alpha}$$
$$\overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2} \times \overrightarrow{D_\beta}$$
$$\overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3} \times \overrightarrow{D_\delta}$$
$$\vec{X}(t+1) = \left(\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}\right)/3$$

where $\overrightarrow{C_1}, \overrightarrow{C_1}$, and $\overrightarrow{C_1}$ are random vectors obtained by Eq. 10 and $\overrightarrow{A_1}, \overrightarrow{A_2}$ and $\overrightarrow{A_3}$ are calculated. This shortcoming is most noticeable in the first stages of development when a greater quantity of exploratory power is required.

Experimental Results

The QoS parameters and agent-device communication enable performance assessment in this section. Agent and device throughput, as well as the total amount of sent and received messages, may all be calculated using the aforementioned metrics. A GWO algorithm is indeed an advisor approach to ensuring premium service is consistently provided. Quality of Service factors taken into account:

- Response Time: Duration of a service's response time to a customer inquiry
- Availability: Time the service is really live as a percentage
- Throughput: The highest rate at which requests may be fulfilled in a particular time interval.

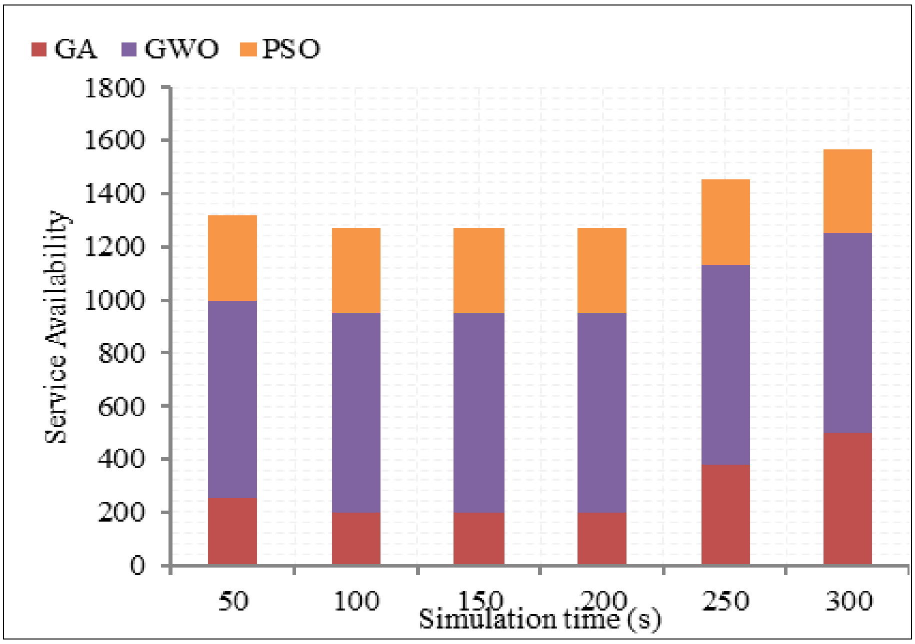

Fig 2: Comparative analysis on packets delivered

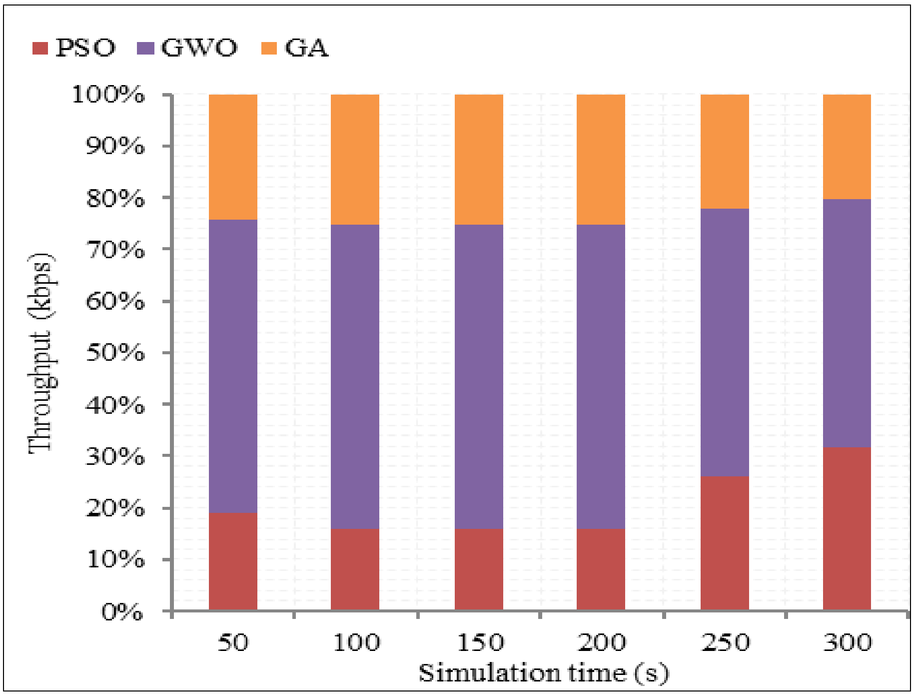

Fig 3: Comparative analysis on throughput

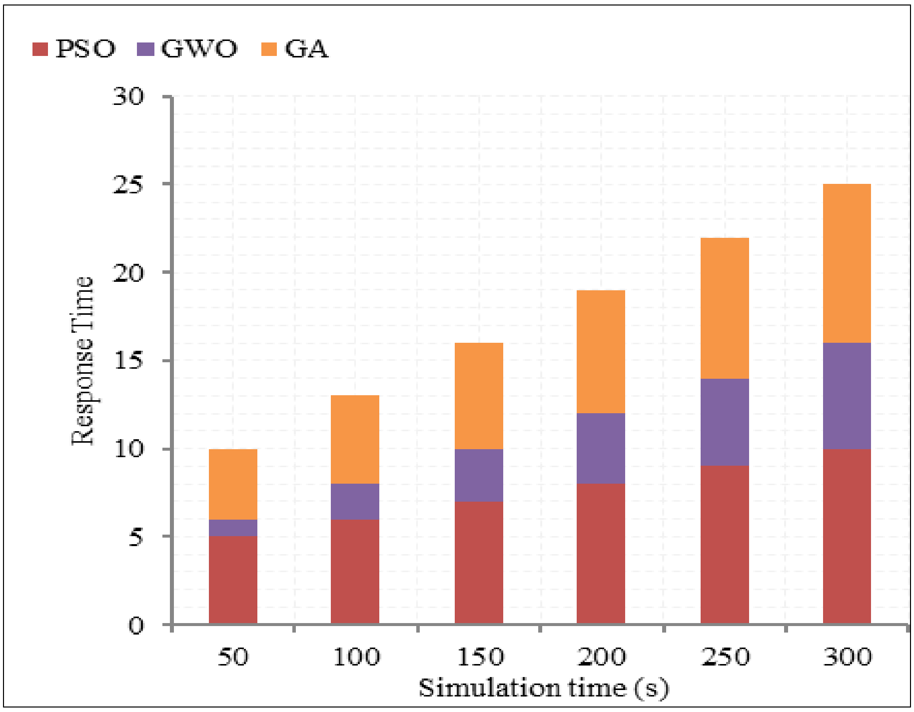

Fig 4: Comparative analysis on Response Time

In this paper, we explored a novel method for finding online services, one that makes use of agents. Via registration, verification, certification, and confirmation of QoS, the agent aims to ease the process of discovering new online services. The representative is accountable for the selection and actual dissemination of online services. We outlined the primary aspects of agents whose needs are not being satisfied by existing approaches to addressing quality of service in the digital realm. We provide an agent-based approach to interactive online service discovery that hides the system's complexities from consumers. Instead of creating a brand new agent from scratch, the service provider may choose to use an existing one. The agent services also provide a great deal of assistance in discovering other online services that may be of use to the user. The quality-of-service characteristics upon which our theoretical architecture is based and is put into practise. A minimum number of service calls must be fulfilled before the system's performance can be evaluated. This will allow for a framework that is both flexible and sturdy.

Conclusion

The study's goal is to use an agent-based strategy to solve the worldwide and decentralized service selection issue. When choosing QoS using an agent-based strategy, agents only exchange information with one another on a local level. In this application, we use the optimization algorithm, which engages in rapid message exchange (namely, UTIL and VALUE messages). QoS parameters, such as both static and dynamic characteristics needed by devices, are used to determine the optimum service to use. Finding the greatest service in a short amount of time is a huge time saver. This agent-based method of selecting the most appropriate web service for a given device is quite helpful. In the long run, we want to implement a combined strategy of local selection methods and global optimization. Then, we utilize mixed integer programming (MIP) and determine the best approach for breaking down global QoS requirements into more manageable, local ones. Secondly, we use distributed localized selection to identify the most suitable Web services that meet these regional requirements.

References

1. Siebert LC, Aoki AR, Lambert-Torres G, Lambert-de-Andrade N, Paterakis NG. An agent-based approach for the planning of distribution grids as a socio-technical system. Energies; c2020.
2. Manukonda K. Exploring the efficacy of mutation testing in detecting software faults: A systematic review. International Journal of Computer and Artificial Intelligence. 2020;7:71-77. DOI:10.5281/zenodo.11408273.
3. Yenugula M, Kodam R, He D. Multiple data centers intended for latency minimization using artificial intelligence algorithms. International Journal of Computer and Artificial Intelligence. 2020;1(1):39-45. DOI:10.33545/27076571.2020.v1.i1a.79.
4. Manukonda K. Efficient test case generation using combinatorial test design: Towards enhanced testing effectiveness and resource utilization. European Journal of Advances in Engineering and Technology. 2020;7(12):78-83.
5. Yenugula M. Data center power management using neural network. International Journal of Advanced Academic Studies. 2021;3(1):320-325. DOI:10.33545/27068919.2021.v3.i1d.1124.
6. Yenugula M, Kodam R, He D. Performance and load testing: Tools and challenges. International Journal of Engineering and Computer Science. 2019;1(1):57-62. DOI:10.33545/26633582.2019.v1.i1a.102.
7. Zhang Y, Meng K, Luo F, Yang H, Zhu J, Dong ZY. Multi-agent-based voltage regulation scheme for high photovoltaic-penetrated active distribution networks using battery energy storage systems. IEEE Access. 2020;8:7323-7333.
8. Castillo-Cagigal M, Matallanas E, Caamaño-Martín E, Martín ÁG. SwarmGrid: Demand-side management with distributed energy resources based on multifrequency agent coordination. Energies; c2018.
9. García-Magariño I, Lacuesta R, Lloret J. ABS-SmartComAgri: An agent-based simulator of smart communication protocols in wireless sensor networks for debugging in precision agriculture. Sensors. 2018;18(11):1-18.
10. Manukonda K. Performance evaluation and optimization of switched Ethernet services in modern networking environments; c2020.
11. Manukonda KR. Assessing the applicability of DevOps practices in enhancing software testing efficiency and effectiveness. Journal of Mathematical and Computer Applications; c2020.
12. Vanrompay Y, Rigole P, Berbers Y. Genetic algorithm-based optimization of service composition and deployment. In: Proceedings of the 3rd International Workshop on Services Integration in Pervasive Environments; New York, NY; c2008. p. 13-18.
13. Wu X, Xiong X, Ying J, Yu C. QoS-driven global optimization approach for large-scale web services composition. Journal of Computers. 2011;6(7):1452-1460.
14. Sasikaladevi N, Arockiam L. Genetic approach for service selection problem in composite web service. International Journal of Computer Applications. 2012;44(4):22-29.
15. Yang W, *et al*. A hybrid particle swarm optimization algorithm for service selection problem in the cloud. International Journal of Grid and Distributed Computing. 2014;7(4):1-10.