

Improving Service Selection for Cloud Service Brokerage using Hybrid Method

Faruku Umar Ambursa
Information Technology Department,
Bayero University Kano.

ABSTRACT

In today's landscape, scheduling users' applications on distributed web services has become increasingly challenging due to the diverse and stringent nature of users' requirements. Additionally, the web market is flooded with a large number of web services, many of which exhibit dynamic performance characteristics during runtime, leading to significant uncertainties. Consequently, service agents face difficulties in selecting the most suitable web services that can meet the users' Quality of Service (QoS) demands. As a result, users often experience violations of their requirements. Several mechanisms have been proposed to address these issues. However, in demanding scenarios, these mechanisms often generate subpar solutions of low quality. To tackle this problem, this paper introduces a service selection scheme specifically designed for cloud service brokers. This scheme aims to minimize overhead while efficiently searching and recommending a suitable set of services that can fulfill the users' QoS demands. Experimental results demonstrate the capabilities of the proposed strategy and its effectiveness in producing satisfactory outcomes.

ARTICLE INFO

Article History
Received: January, 2023
Received in revised form: February, 2023
Accepted: April, 2023
Published online: June, 2023

KEYWORDS

Service selection, cloud service broker,
Artificial Intelligence, Particle Swarm
Optimization

INTRODUCTION

Recently, the Web has experienced high proliferation of cloud services. The key players in the cloud service provider marketplace include AWS, Google cloud, Microsoft Azure etc. However, enormous specialist providers, who offer computing and storage platforms as well as other hosting services, exist. Cloud marketplace is of two types: local, which is formed by services offered by a single provider such as AWS marketplace; or global, which offers services from different participating providers. Moreover, there are huge varieties of cloud services, particularly in the global marketplace, and each variety can be offered by several providers. Additionally, multiple cloud service providers can possess similar quality of service (QoS) attributes. Moreover, it is a common requirement of many industry projects to combine capability and capacity from multiple cloud service providers. However, determining which service provider to include in the service composition requires assessment and evaluation of every provider, which is evidently a serious challenge on ordinary customers. Cloud service brokers are usually employed for this purpose. A cloud broker is "an entity that manages the use, performance and delivery of cloud services, and negotiates relationships between cloud providers and cloud consumers" [1]. It is typically an organisation that

acts as an intermediary between subscribers/end-users and SaaS vendors [2]. However, performance of a services broker is affected by the kind of algorithms used in their service selection and composition policy. Due to the difficulty of the services selection problem, intelligent optimization algorithms, such as genetic algorithm (GA), ant colony optimization (ACO), particle swarm optimization [3], are common in broker design. Nevertheless, the high diversity and stringency of users' requirements, couple with the huge number of web services with their performance uncertainties, has made it difficult for service agents to meet Quality of Service (QoS) demand of users. In consequence, users' requirements are often violated. Various mechanisms have been proposed on these issues. However, in stringent situations, the mechanisms often produce low-quality solutions. Schemes have been proposed in the literature to address this interesting issues. One of the prominent algorithms under this category [4-6] is the multi QoS workflow scheduling strategy based on particle swarm optimization and a look ahead heuristic (LAPSO) [7]. LAPSO was proposed to improve performance and handle complex user requirements. The algorithm selects the best scheduling solutions based on a novel constraint-handling strategy. It hybridizes PSO with a novel look-ahead mechanism based on a min-max

Corresponding author: Faruku Umar Ambursa

✉ fuambursa.it@buk.edu.ng

Information Technology Department, Bayero University, Kano, Nigeria
© 2023. Faculty of Tech. Education, ATBU Bauchi. All rights reserved

heuristic, designed to improve quality of solutions. At the core of the LAPSO operation is the Look-Ahead Task-Remapping algorithm (LATR), which improves the best solution found by PSO at end of every iteration of the algorithm. However, the LATR waste CPU resource in its attempts to improve incumbent solution.

In this paper, an enhanced LATR (ELATR) is proposed to address the aforementioned issue. The proposed method is based on a Tournament selection technique. It aims to offer the advantages of: ensuring minimum/no violation in QoS delivery to stringent user demands; and improving user satisfaction in the QoS rendered. Results of experiments demonstrate capabilities of the proposed strategy.

In the remainder of this paper, Section 2 discusses related works. The operations of the LAPSO, LATR as well as the proposed Tournament-based selection strategy are presented in Section 3. Section 4 highlights the performance results. Finally, Section 5 concludes the paper and suggests future work.

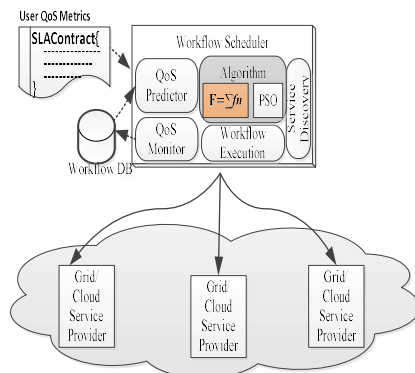


Figure 1: Reference architecture

RELATED WORKS

The workflow scheduling in distributed computing resources has been a topic of interest for many researchers. A lot of works address the scheduling of single workflow. Few types of research consider multiple workflow scheduling but majority of them consider limited user quality of service (QoS) requirement.

In [6] the authors present two online algorithms to schedule multiple workflows under deadline and privacy constraints while considering the dynamic nature of hybrid cloud environment. Elastic Budget-constrained

resource Provisioning and Scheduling algorithm for multiple workflows was proposed in [7] for WaaS a platform. It was claimed that the method reduces the computational overhead by encouraging resource-sharing policy to minimize workflows' makespan while meeting user-defined budget. In [8], a look ahead algorithm is proposed to solve online multiple workflows scheduling problem. Reference [9] proposed a competent dynamic multi workflow scheduling (CMWS) algorithms which are divided into two phases to reduce the system overhead and power consumption. Fairness Scheduling with Dynamic Priority for Multi Workflow (FSDP) was proposed in [9]. FSDP emphasized the fairness aspect as it integrated slowdown metrics to their algorithm's policy. Slowdown value was the ratio of the makespan of a workflow when it was being scheduled in dedicated service to the makespan of it being scheduled in a shared environment with the other workflows. The authors in [10] proposed a fairness-based scheduling algorithm called fairness-based dynamic multiple heterogeneous selection values to achieve high performance of systems and a priority-based scheduling algorithm called priority-based dynamic multiple heterogeneous selection value. They combine these algorithms to presents a tradeoff based scheduling algorithm to meet the deadlines of more higher-priority workflows while still allowing the lower-priority workflows to be processed actively for better performance of their systems. A multi-objective optimization workflow scheduling approach was presented based on dynamic game-theoretic model [11]. The approach aims at reducing workflow make-spans and total cost while maximizing system fairness in terms of workload distribution among heterogeneous cloud virtual machines (VMs). In [12] the authors proposed an approach for scheduling multiple workflows that takes fairness into account with three different Qos objectives and four workflow selection policies based on unfairness metric: In ordered policy, round robin, weighed round robin, and direct heuristic policy. However, these approach takes into account very limited Qos objectives and therefore difficult to fit well in the present dynamic and complex cloud environment.

In [7] a constraints aware multi Qos workflow scheduling strategy based on particle swarm optimization and a look ahead heuristic (LAPSO) was proposed to improve performance

and handle complex user requirements. The algorithm selects the best scheduling solutions based on a novel constraint-handling strategy. It hybridizes PSO with a novel look-ahead mechanism based on a min-max heuristic, designed to improve quality of solutions. The Multi-QoS scheduling model is made up of six QoS constraint time, cost, reliability, availability, security, and reputations. To select services set for a workflow, two models were used: cumulative QoS violation rate, which defines violation degree of user defined constraints; and cumulative QoS satisfaction rate, which is defined by the weighted sum of the values of the six QoS objectives. At the core of the LAPSO operation is the Look-Ahead Task-Remapping algorithm (LATR), which improves the best solution found by PSO at end of every iteration of the algorithm. The LAPSO's LATR algorithm achieves its purpose by making two attempts: in the first attempt it uses the best solution of the last iteration to improve the best solution of the current iteration; if no improvement is achieved in the first attempt, the algorithm makes second attempt, where, for the sake of simplicity, an arbitrary solution is randomly choosing from the population. However, a closer study of the algorithm reveals that the arbitrary selection of a solution from the solution space leads to numerous failed attempts. It is believed that addressing the failure rate will significantly boost the algorithm performance.

In the next section, the LATR of the LAPSO algorithm is extended to minimize the failure rate.

Service Selection Method for Cloud Service Brokerage

This section also demonstrates a proposed mechanism for refining scheduling solutions. In what follow, the two aspects of the study are discussed separate subsections.

Overview of the LAPSO Algorithm

The algorithm is based on PSO technique. It operates iteratively in three logical phases: solutions search, solutions evaluation and selection and solution improvement. The search phase explores the available services and returns a number of options called candidate solutions. In the evaluation phase, the solutions are evaluated based on QoS violation level and total QoS value and a single best solution is selected. In the improvement phase, the scheduler checks for any service of the best

solution with lower QoS value and replace it with a better service. Finally, the improved best solution is returned to service agent if it fulfils the user's requirements, else the scheduler goes to search phase again for the next iteration. The reader is referred to [7] for details of the LAPSO algorithm. The algorithm uses a novel LATR algorithm for the refinement of the best solution found by PSO at the end of each iteration.

Algorithm 1: Improved Look-ahead QoS-aware Task-Remap

```

Input:  $\phi_{best}^f$ 
Output:  $\phi_{best}^+$ 
1   $\sigma_{inferior} \leftarrow Search()_{QoSSet_{best}}$ 
2   $\phi_f^{\sigma_{inferior}} \leftarrow Search(\sigma_{inferior})_{MapSet_{best}}$ 
3  counter  $\leftarrow 1$  //first attempt to improve  $\phi_f^{\sigma_{inferior}}$ 
4   $\phi_{determinant} \leftarrow \phi_{best}^{i-1}$ 
5   $\phi_f^{\sigma_{inferior}} \leftarrow Search(\sigma_{inferior})_{MapSet_{determinant}}$ 
6  If  $\phi_f^{\sigma_{inferior}} > \phi_f^{\sigma_{inferior}}$  then
   | Line 7 through 18, the algorithm tests whether remapping
   | the task to the service of the determinant solution can
   | maximise the overall QoS of the best solution [4].
19 Else
20   If counter  $\leftarrow 1$  then
21     counter ++ //second attempt to improve  $\phi_f^{\sigma_{inferior}}$ 
22     //invoke Algorithm 2
23      $\phi_{determinant} \leftarrow tournament(Size, curr\ PSO\ swarm)$ 
24     Go to line 5
   Return  $\phi_{best}^+$ 

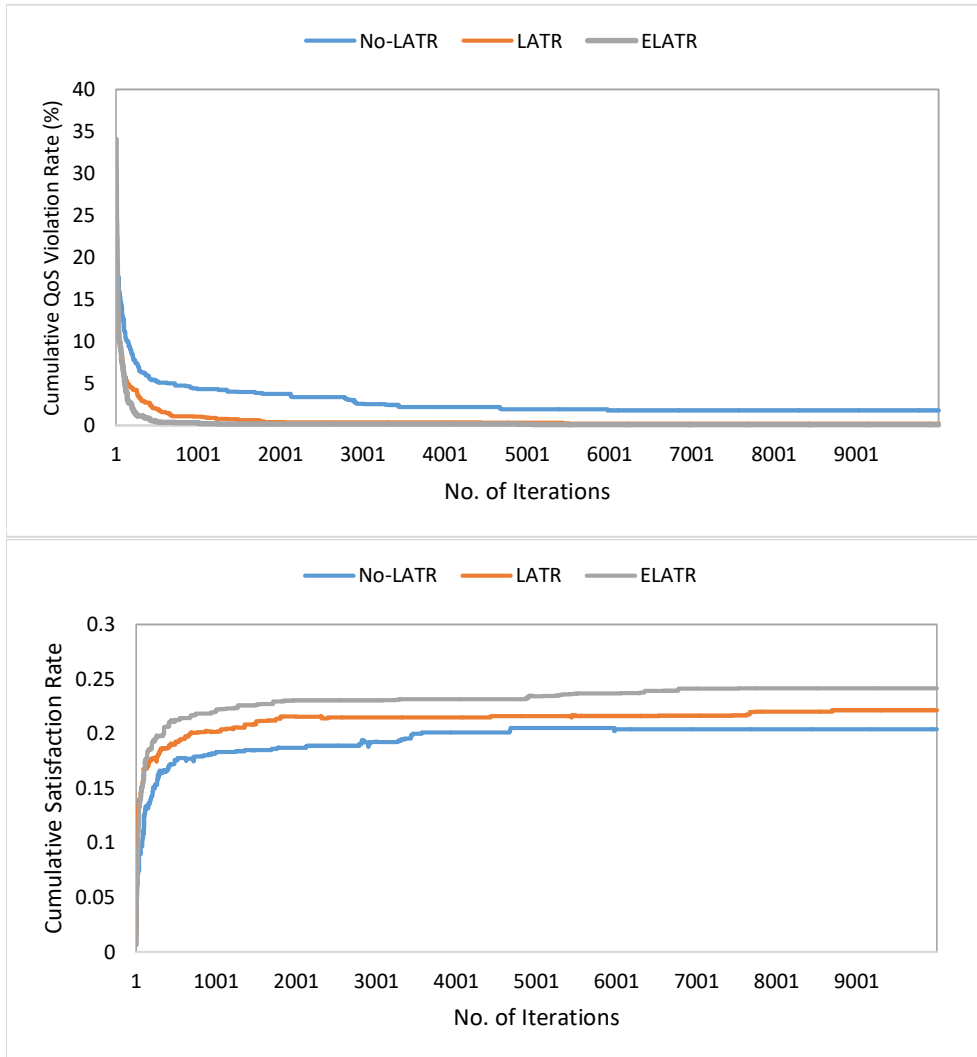
```

Overview of the Look-Ahead Task-Remapping (LATR) Technique

The LATR technique refines the best solution of each PSO iteration by deterministically altering one of the task-service mappings of the solution. It chooses the task with the lowest parameter value and re-assign it to an alternate service that potentially improves its value. The LTR algorithm learns from the good mappings in other solution, called determinant scheduling solution $\phi_{determinant}$, to make changes in the best solution. The $\phi_{determinant}$ serves as a guide for the remapping. To find the $\phi_{determinant}$ the algorithm makes two attempts: in the first attempt, the best solution of the previous iteration is used; in the second attempt, the $\phi_{determinant}$ is selected randomly from the solution space as follows:

$$\phi_{determinant} \leftarrow rand(current\ PSO\ swarm) \quad (1)$$

The quality of the chosen arbitrary solution is then tested in terms of value of the target parameter as well as potential for improving



(line 5). Line 6 through 18, the algorithm tests whether remapping the task to the service of the determinant solution can maximise the overall QoS of the best solution [4]. If however the condition of line 6 is not satisfied, then the algorithm makes a second attempt by invoking Algorithm 2, test and possibly effect the output of the tournament-based method and then returns the refined best solution as output (line 19-24).

EXPERIMENT AND RESULTS

Experimental Setup

This section presents the experiments environment for the proposed method such as computer resources, network topologies and the

experimental setup. The resources are used to test the performance of the proposed algorithm. The experimentation study was conducted by means of simulation using Eclipse Java Programming Environments. The simulation was performed on a computer with a Pentium (R) Dual- Core CPU (2.30 GHz, 2 Cores) and 8GB RAM. The operating System was MS Windows 10. The algorithms were tested using a e-Protein workflow application (shown in Fig. 2) as used in the benchmark work [4]. The performance of the proposed method is evaluated using two metrics: percentage satisfaction rate and percentage violation rate [4]. The performance of the method is compared with the LATR, which is random-

Corresponding author: Faruku Umar Ambursa

fuambursa.it@buk.edu.ng

Information Technology Department, Bayero University, Kano, Nigeria

© 2023. Faculty of Tech. Education, ATBU Bauchi. All rights reserved

based and the LAPSO without LATR (original). The algorithms were tested using a e-protein workflow application with 15 tasks, the DAG of which is depicted in Fig. 2, as used in reference [xx]. The workflow tasks were assigned each a set of service instances ranging from 60 to 120. Refer to [4] for details of the service instances. As in the previous work, six QoS parameters were used and tighter QoS requirements were set, viz: deadline, budget, reliability, availability, security and reputation requirements were set to 22.0, 70.0, 0.85, 0.85, 5.6 and 4.7 respectively.

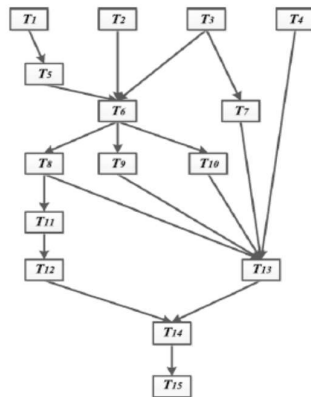


Figure 2: e-Business workflow [4]

Result

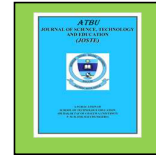
Fig. 3a and 3b depicts the experimental results of the three compared approaches in terms of percentage QoS violation rate and QoS satisfaction rate, respectively. From the simulation results, it can be observed in Fig. 3a that enhancement of LATR (ELATR) achieves best performance compared to LATR and, of course, the No-LATR. No-LATR achieves lowest performance because it does not improve best solution found by PSO. Specifically, the ELATR achieves lowest violation of only 0.11, while LATR recorded 0.2. Both LATR and ELATR shows good performance, but ELTR is better. The No-LATR recorded about 2% violation. On the other hand, Fig. 3b shows performance of the compared algorithms in respect of QoS satisfaction rate. From the figure, again the proposed ELATR outperformed the compared. Specifically, the proposed method recorded 0.24 and LATR achieved 0.22, whereas No-LATR got 0.20. All the algorithms achieved good performance, but ELATR performance is better.

CONCLUSION

In this study, an enhanced LATR algorithm is presented. LATR algorithm is an auxiliary mechanism that assists LAPSO in refining best solutions at end of each PSO iteration. The ELATR is based on the Tournament Selection method, which selects a set of individuals for a tournament, where the best fit individual emerges as the winner. This addresses the problem of selecting only a single individual randomly from the PSO population. Results of experiment shows that although LATR is good, ELATR achieved better performance. In future, the ELATR will be tested with other evolutionary techniques.

REFERENCES

- [1] Y. Wang, J. Jiang, Y. Xia, Q. Wu and X. Luo, "A Multi-stage Dynamic Game-Theoretic," Springer International Publishing AG, part of Springer, pp. 137-152, 2018.
- [2] X. Zhenzhen, C. Xin, X. Xiujuan, Z. Xiaowei, J. J. Dai and Mingfei, "A look-ahead algorithm for online multiple workflow scheduling problem," *Concurrent Engineering: Research*, 2017.
- [3] M. H. Hilman, M. A. Rodriguez and R. Buyya, "Resource-sharing Policy in Multi-tenant Scientific Workflow as a Service Platform," *JOURNAL OF LATEX CLASS FILES*, VOL. 14, NO. 8., 2015.
- [4] Singh, G., & Chaturvedi, A. K. (2021, May). Particle swarm optimization-based approaches for cloud-based task and workflow scheduling: a systematic literature review. In *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)* (pp. 350-358). IEEE.
- [5] Saif, F. A., Latip, R., Derahman, M. N., & Alwan, A. A. (2022, October). Hybrid meta-heuristic genetic algorithm: Differential evolution algorithms for scientific workflow scheduling in heterogeneous cloud environment. In *Proceedings of the Future Technologies Conference* (pp. 16-43). Cham: Springer International Publishing.
- [6] Alsaih, M. A., Latip, R., Abdullah, A., Subramaniam, S. K., & Ali Alezabi, K. (2020). Dynamic job scheduling strategy using jobs characteristics in cloud computing. *Symmetry*, 12(10), 1638.



- [7] U. A. Faruku, L. Rohaya, A. Azizol and S. Shamala, "A particle swarm optimization and min-max-based workflow scheduling algorithm with QoS satisfaction for service-oriented grids," Springer Science+Business Media New York 2017, 2017.
- [8] M. Adhikari and S. Koley, "A Multi-workflow Scheduling Algorithm with Dynamic Reusability," *Arabian Journal for Science and Engineering*, 2017.
- [9] Y. Wang, S. Cao, Z. Feng, G. Wang and H. G. Zhang, "Fairness Scheduling with Dynamic Priority for Multi Workflow on Heterogeneous," in 2017 the 2nd IEEE International Conference on Cloud Computing and Big Data Analysis, 2017.
- [10] R. Amin, N. Mahmoud and E. D. H. J., "Fair multiple-workflow scheduling with different quality-of-service goals," Springer Science+Business Media, LLC, part of Springer Nature 2018, 2018.
- [11] S. Sharif, J. Taheri and A. Y. Zomaya, "Online Multiple Workflow Scheduling under Privacy and Deadline in Hybrid Cloud Environment," 2014 IEEE 6th International Conference on Cloud Computing Technology and Science, 2014.
- [12] G. Xie, G. Ze ng, J. Jiang, C. Fand, R. Lia and K. Li, "Energy Management for Multiple Real-time Workflows on Cyber-Physical Cloud Systems," *Future Generation Computer Systems*, 2017.
- [13] Bäck, T., Fogel, D. B., & Michalewicz, Z. (Eds.). (2018). *Evolutionary computation 1: Basic algorithms and operators*. CRC press.