INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 17, Issue: 2, Month: April, Year: 2022
Article Number: 4296, https://doi.org/10.15837/ijccc.2022.2.4296

A Unique Multi-Agent-Based Approach for Enhanced QoS Resource Allocation in Multi Cloud Environment while Maintaining Minimized Energy and Maximize Revenue

Umamageswaran Jambulingam, K. Balasubadra

Umamageswaran Jambulingam
Department of Information Technology
R.M.K. Engineering college, RSM Nagar, Kavaraipettai, Thiruvallur Distric, Tamil Nadu, 601206.
Corresponding author:umamageswaranphd@yahoo.com

K. Balasubadra
Department of Computer Science and Engineering
R.M.D. Engineering College, RSM Nagar, Kavaraipettai, Thiruvallur Distric, Tamil Nadu, 601206.
balasubadra@yahoo.com

Abstract

The use of the multi-cloud data storage in one heterogeneous service is a polynimbus cloud strategy. Cloud computing uses a pay-as-you-go model to deliver services to a variety of end users. Customers can outsource daunting tasks to cloud data centres for processing and producing results, thanks to cloud computing. Cloud computing becomes the popular IT brand that provides various on-demand services over the internet. This technology is devoted to distributing computer and software resources. The proven usefulness of workflows to enforce relevant scientific achievements is the availability of data from advanced scientific tools. Scheduling algorithms are essential in order to automate these strenuous workflows efficiently. A number of new heuristics based on a Cloud resource model have been developed. The majority of these heuristic - based address QoS issues in one or two dimensions. The cloud computing technology offers a decentralised pool of services and resources with various models that are provided to the customers across the Internet in an on-demand, continuously distributed, and pay-per-use model. The key challenge we address in this paper is to maximise revenue while maintaining a minimum consumption of energy with an enhanced QoS for resource allocation. The obtained results from proposed method when compared with the existing state of art methods observed to be novel and better.

Keywords: Artificial Bee Colony (ABC), Best Fit Decreasing (BFD), Distributed Energy Resources (DER), Economic Dispatch (ED), Genetic Algorithm (GA), Multi Agent System (MAS), Priority based Resource Allocation (PRA), Service-Level Agreement (SLA), Vickrey –Clarke–Groves (VCG), Virtual Machine (VM).

1 Introduction

The world is witnessing rapid developments in software that is used on a daily basis. Users' demands for more services are constantly growing as new online services arise. As a result, the task of ensuring a high level of service is at the center of business competition. New technologies, especially cloud computing, give businesses the opportunity to provide strong mobile services at a low cost. Customers who use cloud computing are relieved of the responsibility of owning and running the physical infrastructure that their companies need. This eliminates the need for them to be concerned with developers and development teams. Customers often don't have to worry about how or when the necessary tasks are completed; all they have to worry about is the cost of using the technology, the services they will receive from cloud providers, and the service quality promises. Cloud providers are concerned with achieving effective resource usage and provisioning using intelligent ways to track and control resources, and using various approaches to achieve the highest quality of service assurances without violating the SLA. These are a few of the major obstacles that cloud providers face. As the Internet grows in popularity, so does the number of computational techniques available. A growing volume of data must be processed in this situation. The introduction of various types of cloud computing, such as cloud computing, edge computing, and fog computing is due to the increase in user requirements.

2 Generic related work

Cloud computing is a pioneering virtualization technology that has greatly aided data processing. It enables network access to system configurable resources such as networks and the internet in a simple and fast manner. Furthermore, provisioning and publishing these services do not necessitate a lot of management or service provider contact [1]. Figure 1 depicts the cloud computing framework.

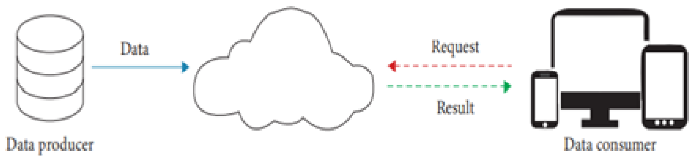

Figure 1: : Structure of Cloud Computing

The IoT framework cloud computing technology faces some limitations as the Internet of Things evolves and people's needs grow. Cloud computing cannot play a useful role in large-scale or homogenous situations in this case [2]. As a result, a new computing paradigm based on cloud computing called fog computing is being created. The key benefit of fog computing over cloud computing is that it extends cloud services to the network edge. As a result, fog computing can help with resource and service management [3]. Figure 2 depicts the fog computing structure. Edge computing enables operations to be carried out at the network's edge [4]. Edge computing encompasses all computing and network infrastructure, from data sets to cloud data centers. The computing flow is bidirectional in edge computing, and stuff in edge computing can consume and generate the data. They can, in other words, not only request cloud services but also perform computing tasks in the cloud Figure 3 depicts the structure of edge computing. The MEC, which refers to the engineering of completing graphics rendering and postponement tasks for mobile devices, is the most common representation of IoT technology. And its theory entails gathering a significant amount of free processing power and storage capacity at a network's edge. It was first described as a computing model by the European Telecommunication Institutional Structure. MEC provides the capabilities of information and edge hosting at the network edge. Elasticity in cloud computing defined as the degree of automated service discovery adaptation in reaction to continuously changing in the customer's workload and requirements. This is accomplished by scaling up or down the services allocated to a specific customer automatically. Such a system should be as similar to the original as possible. the available capital in relation to the existing customer demands [5]. As a result, elasticity can simply be defined as the absence of both the overprovisioning and the under-provisioning issues resulting in successful resource provisioning [6] If there is an overabundance of supplies, the issue of overprovisioning will arise.

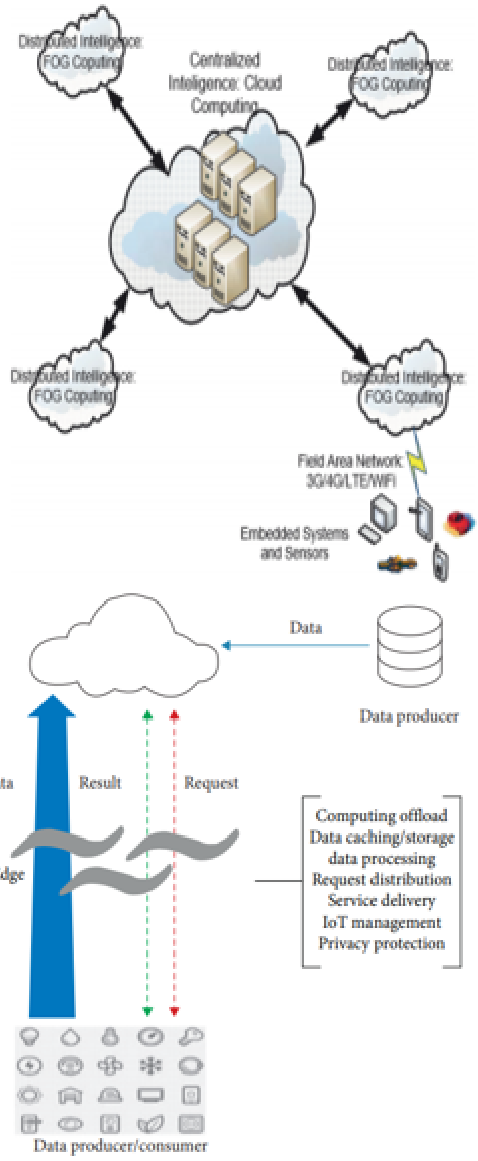

Figure 2: : Structure of Fog and Edge Computing.

A customer's reserved resources are insufficient to meet his or her needs as seen in the left part of Fig. 1. The red line in the diagram represents the available resources based on peak load is calculated right, resulting in no SLA violations. However, without elastic modulus during non-peak times, resources will be depleted are thrown away. The problem of under provisioning, on the other hand, may occur when the reserved assets are insufficient for the current situation customer's requirements This issue results in SLA violations. As a result, sales and clients are lost. As seen in the diagram, the under-provisioning problem can manifest itself in a variety of ways. Sections of Fig. 3 in the center and right 1 in each case. The shaded areas in these graphs reflect the SLA violations. may change over time depending on the needs of the customer.

3 Existing works

Multiagent-Based Resource Allocation for Energy Minimization in Cloud Computing Systems (RAEM):

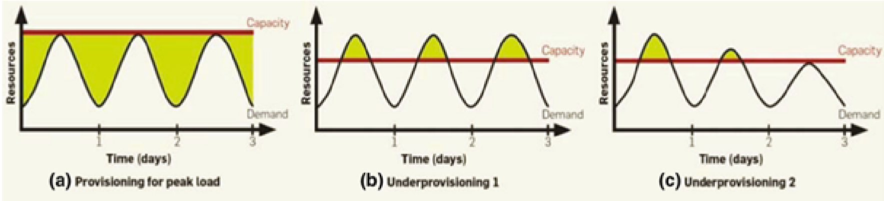

Figure 3: : Overprovisioning and under provisioning

The main advantage of this approach lies at allocation and consolidation of VMs are done very efficiently. A Centralized BFD-GA approach is used which is helpful for VM allocation and VM consolidations. [7] suggested a Vickrey–Clarke–Groves (VCG)-based model based on a truthful mechanism to maximize revenue [8] suggested a traded protocol based on the MA to trade efficiently and effectively fair distribution of resources among selfish users [9] suggested being honest. mechanisms while maximizing the profit of the CSP Many of these market-driven processes densely model the market for convenience's sake. The cost of energy for cloud systems is proportional to the amount of data stored. Users are assigned a certain number of virtual machines (VMs). MA technology, that is developed from centralized artificial intelligence, has demonstrated the ability to solve distributed system issues [10]. The bi-objective parameters which are associated with revenue and cost minimization need to be addressed in a well manner, which is the main pitfall of this approach.

Multi-Agent Systems for Resource Allocation and Scheduling in a Smart Grid (RASG): The major advantage of this method lies at effective scheduling and allocation of resources. A unique integration of DER-ED approach is carried out to prove the desired results. With the proposed method an effective usage of MAS was defined. An agent can be described as a computer network capable of making critical decisions in response to a scenario in order to achieve its goal [11]. Unit Commitment (UC) is a highly complex optimization technique in a smart grid system that regulates the startup and termination of generators to meet demand while keeping performance parameters in mind [12]. Algorithms that can be distributed Such algorithms are stable, resistant to topological changes, and can facilitate future grid's "plug-and-play" functionality. It's a strategy for getting power applications to settle on a single data value. This algorithm is intended for use in networks with many unreliable nodes [13]. [14] suggested a decentralized consensus-based design for the ED in a smart grid system that preserves supply-demand equilibrium during transients. Since the approach does not depend on a supply-demand discrepancy, it could be used online. The main downfall of this approach is related in defining a set of constraint applications related to Multi Agent System.

Priority based Resource Allocation and Scheduling using Artificial Bee Colony (ABC) Optimization for Cloud Computing Systems (PBRA) Efficient resource utilization and allocation is the main contribution of this approach. PRA-ABC with CloudSim on java platform is the used methodology. [15] has proposed a decentralised multiagent (MA) VM allocation method. To save resources, a local compromise VM consolidation process is built to swap the allocated VMs of agents. [16] uses task project duration and load as algorithms when deciding which VMs to use. The obtained results from this approach were satisfiable. The main pitfall of this approach is that it works only for certain predefined parameters.

Optimization Approach for Resource Allocation on Cloud Computing for IoT (OARA): Main highlight of this approach lies at calculating accurately the probability of deadline and provider's profit. Combinatorial auction methodology is used with an eye on deadline of the given job. Advantage of this approach lies at delivering satisfiable results when compared with the existing methods. The big data problem has been recognized as a worldview for cloud computing [17]. An auction-based model is the most popular method in allocation of resources and pricing in the cloud [18]. In cloud computing, combinatorial auction is favored because it allows users to purchase a bundle of resources rather than a single resource. Before a provider could provide a provider to a user, both the provider and the user must agree on a service level agreement (SLA). A service level agreement (SLA) is an agreement between such a provider and a user that specifies QoS [19]. loss of profit for a create a shared to failure to complete certain jobs and the cost of a SLA penalty [20]. We use the probability of dateline violations by considering the job's urgency when determining winners to maximize the provider's profit by lowering the penalty function for SLA violations [21]. The major pit fall of this approach lies at working of this model with certain constrained parameters on the combinatorial auctions.

Author	Contribution	Methodology	Advantage	Limitations
Wanyuan Wang et al. [22]	Efficient allocation of VM, VM Consolidation	Centralized bin BFD and GA approach	Better in VM allocation and consolidation	Bi-objectives of cost minimization and revenue need to be addressed
Arun Sukumaran Nair et al. [23]	Effective allocation and scheduling of resources	Integrated DER-ED	Efficient usage of MAS	Further application of MAS needs to be explored
A Phani Sheetal et al. [24]	Efficient resource utilization and allocation	PRA-ABC with CloudSim	Results are satisfiable compared to existing one	Confined to work with limited parameters
Yeongho choi et al. [25]	Probability of deadline and providers profit is calculated	Combinatorial auction giving importance to job's urgency with constraint in deadline	Obtained results are satisfiable compared to existing one	combinatorial auction works for defined parameters

Table 1: Comparisons of exiting methods

4 Proposed work

In order to address the key challenges associated with delivering QoS while maintaining minimal energy consumption and more profit the following two algorithms are proposed.

Algorithm 1: Energy minimization Algorithm

Let A_m be the m available resources, Sopt be the sources which are opted to do the assigned jobs, hostlist consists of available list of resources. After utilizing the resources, the hostlist is then cleaned. Processor speed along with upper and lower bound are being fixed. By maintaining the above condition, a minimum energy consumed by the resources can be maintained.

5 Experimental setup

This work is indented to measure the real-time performance of the cloud resource scheduling algorithms in terms of Availability, Refusal Rate, Reliability, Scalability, Average Energy Consumption, Average Response Time and Cumulative Energy Cost. The standard cloud scheduling procedure codes are fetched from [26] and the proposed method is coded using VC++ programming language.

A dedicated User Interface (UI) is constructed using Visual Studio IDE [27] to establish communication with the Common Gateway Interface (CGI) [28] which connects to a leased Windows Virtual Private Server from [29]. The Express Windows VPS with dual-core processor, 2GB RAM, 60GB SSD, 50Mbps bandwidth and Free DNS is configured to run standard and proposed cloud resource

Input: *host list* $A_n \in \{\varphi\}$, $P_i \rightarrow \{A_1, A_2, \ldots\ldots A_m)$
Target: S_{opt} *for jobs i=1*

1. *Initialize i= hostlist (j):j+1*
2. *Initialize processor speed* k_p, L_{bound}, *and* U_{bound}
3. *Computer Th for* $S_{opt}(0)$
4. *For j =1: j+1 while*
5. *hostlist[j]+hostlist[j+1]*$\leq$ *&&* $\leq t_0 \leq t_1$
6. *Compute* $S_{opt} = S_{opt}[t_0 - 1] + k\ S_{opt}[t_0]$
7. *T{i} = deadline for computation time of task*
8. *sort*S_T *based on* $A_n \in L_{bound} \leq J_{list} \leq Ubound$
9. *If hostlist[j]= EoL then*
10. *Hostlist.remove(j)*
11. *Else*
12. *Hostlist[j].add(j)*
13. *End while*
14. *End for*
15. *Endif*
16. *Return null list.*

Algorithm 2: Maximum revenue algorithm
Input: t, l_t^r, l_t^0, i_t
Output: Maxrev

1. dp <- {-1} /memoization/
2. fun (t, l_t^r, l_t^0, i_t)
3. if dp[t][l_t^r][l_t^0][i_t]$\neq$1 *then*
4. Return dp [t][l_t^r][l_t^0][i_t]
5. end if
6. If t = τ then
7. dp [t][l_t^r][l_t^0][i_t] = 0
8. return 0
9. end if
10. Maxrev<-0
11. for $r_t < -0$ *to* $\min(C - l_t^r - l_t^0, d_i^r)$ *do*
12. rev <-0
13. $l_{t+1}^r < -l_t^r + r_t - e_t^r$
14. $o_t < -\min(C - l_t^r - l_t^0, d_i^0)$
15. $s_t < -\min\ (C - (l_t^r - r_t)z_t - l_t^0 - o_t, d_i^s$
16. λ<-(τ-t)/ τ
17. $\Gamma(\varsigma_t, r_t) < -B\lambda r_t \Phi = BT(\alpha p(l_t^r + r_t)z_t + p(l_t^0 + o_t) + \beta p s_t$
18. for $l_{t+1}^0 < -0$ *to* $l_t^0 + o_t$ *do*
19. for $i_{i+1} < -0$ *to* $|Z|$ *do*
20. $P(\varsigma_{t+1}|\varsigma_t, r_t) < -P\ (u_{t+1} = Z_{i+1})\ x\ Bin(l_t^0 + o_t - l_{t+1}^0; l_t^0 + o_t, q)$
21. $rev < -rev + \gamma(\varsigma_t, r_t) + P(\varsigma_{t+1}|\varsigma_t, r_t)\ x\ V(t+1, l_{t+1}^r, l_{t+1}^0, i_{i+1})$
22. End for
23. End for
24. End for
25. If rev >= maxrev then
26. Maxrev<-rev
27. end if
28. End for
29. dp<- [l_t^r][l_t^0][i_t] $< -maxrev$
30. return maxrev
31. End fun

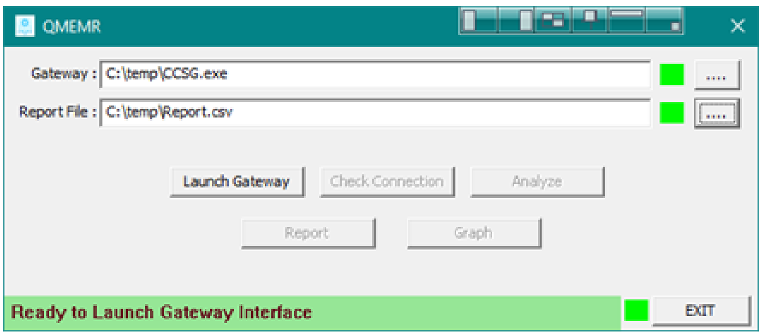

Figure 4: : The User interface

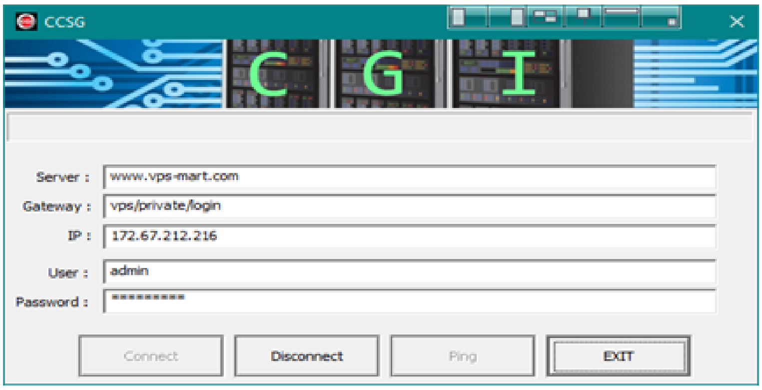

Figure 5: : The CGI data

allocation procedures. The performance monitor is used to measure the required performance parameters and grabbed through CGI to the dedicated UI. Then the UI can export the measured results as the report file and can plot corresponding graphs. To provide QoS resource allocation with minimal

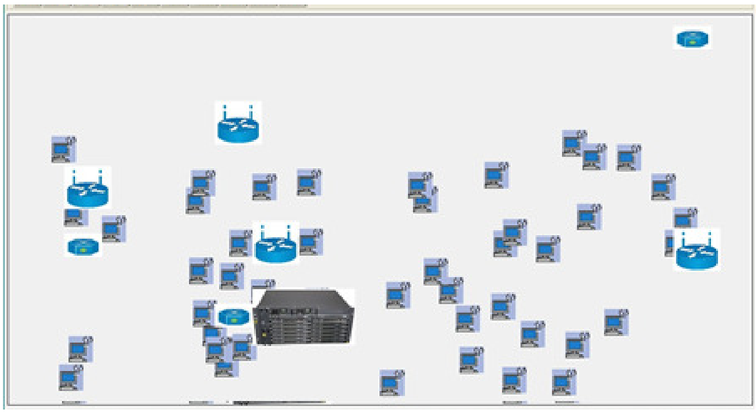

Figure 6: : snapshot of the simulation with multiple agents working in mulit cloud environment

energy consumption and to generate maximum income in accordance with the suggested methodology, all of the VMs indicated above operate as multiagents. Multi-cloud VMs are added to the hotlist and may be used to meet various criteria, including the fixing of different processor speeds in accordance with the need for minimising energy use. Reserved seat contracts that were accepted prior to time zero are used to determine the starting value for an equation such as this t=0. This algorithm's goal is to enable as many reservation agreements as feasible by filling up the nodes from the very end of the window to the very beginning for the maximum profit.

6 Results and discussions

The proposed method was carried out with respect to the parameters availability, refusal rate, reliability, scalability, average energy, average response time and cumulative energy cost. Availability: Represents the availability of resources with more quality of services.

Higher the values of availability indicate the more availability of resources without any conflict, which is clearly achieved from the proposed method and the maximum reached values is nearly 99.02%.

Refusal rate: It indicates how many requests are rejected to the total number of submitted requests and how many are accepted. It is calculated by the following formulae

The less the refusal rate the more will be strength of the proposed method. From the above values in the table, it is clearly observed that the proposed methods have very less refusal rates indicating that the proposed method accepts a greater number of request than rejected ones.

Availability (%)					
Time	**RAEM**	**RASG**	**PBRA**	**OARA**	**QMEMR**
5	90.63	91.77	93.98	96.25	98.93
10	90.28	91.51	93.48	96.12	98.25
15	90.27	90.2	94.23	96.41	98.11
20	90.18	90.83	93.82	96.39	99.17
25	89.25	90.4	93.39	96.11	98.02
30	90.64	91.16	93	97.08	99.42
35	90.96	91.95	93.52	96.61	98.88
40	89.15	90.09	93.22	96.97	98.59
45	89.28	91.07	94.77	96.93	97.82
50	89.5	90.83	94.17	97.94	99.02

Table 2: Availability

Figure 7: : Availability graph for existing and proposed method

$$\boldsymbol{Refusal\ rate} = \frac{Total\ number\ of\ request\ rejected}{Total\ number\ of\ requests\ submited}$$

Refusal Rate (%)					
Time	**RAEM**	**RASG**	**PBRA**	**OARA**	**QMEMR**
5	4.05	3.86	2.91	2.27	1.9
10	4.03	3.8	2.85	2.35	1.92
15	4.08	3.88	2.89	2.25	1.89
20	4	3.95	2.89	2.24	1.83
25	4.14	3.84	2.92	2.26	1.92
30	4.06	3.94	2.81	2.26	1.9
35	4.14	3.91	2.9	2.25	1.85
40	4.09	3.93	2.91	2.28	1.85
45	4.11	3.91	2.8	2.21	1.8
50	4.12	3.91	2.89	2.25	1.81

Table 3: Refusal rate

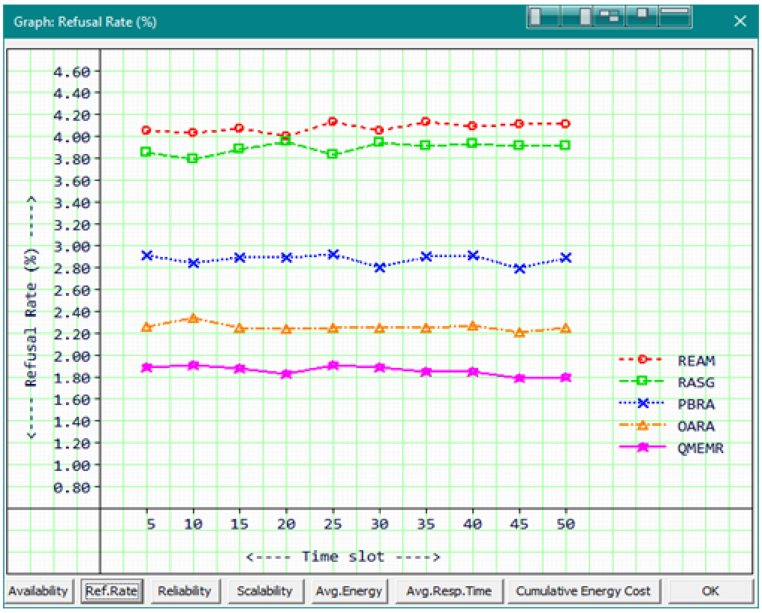

Figure 8: : Refusal rate graph for the existing and proposed method

Reliability: It defines that how reliable is system or resource is working in a cloud environment according to the requirements. Is obtained by the formulae

$$\textbf{\textit{reliability}} \quad R = \prod_{i=1}^{n} \prod_{j=1}^{m} 1 - (1 - R_{C_i})^j$$

Reliability (%)					
Time	**RAEM**	**RASG**	**PBRA**	**OARA**	**QMEMR**
5	89.48	90.54	94.17	96.07	98.86
10	89.04	91.94	94.28	96.4	98.66
15	89.37	90.82	94.04	97.94	97.15
20	89.24	91.5	93.62	97.29	97.04
25	90.75	91.47	94.81	97.15	97.49
30	90.65	91	94.26	97.2	98.93
35	90.83	91.01	93.56	97.65	97.02
40	89.37	90.81	93.51	96.41	98.85
45	90.17	91.22	93.03	97.79	97.4
50	90.54	90.88	94.64	96.86	97.11

Table 4: : Reliability

Higher reliability is more favorable than lower because the resource which has been allocated for a specific purpose in cloud environment should carried out its functionality as per the requirement. It is clearly observed from the above values that the proposed algorithm is helpful in order to make the resources to work according to the need. The highest reliability of the approach obtained is 98.93

Scalability: When and where required depending on the need the resources which are utilized in cloud environment should be scalable

From the above table it is clear that higher the scalability the more will be the strong approach. From the obtained values of the proposed method, it is very clear that our approach is more scalable with respect to resources. Where the maximum scalability is 99.4%.

Average Energy Consumption: or a specific task in cloud environment, it indicates the average energy consumed by the individual resources which are dedicated for a specific purpose.
response rate=current time-arrival time+remaining time

The less the Average Response Time the more will be the effort of the system. Form the values in the table it is clear that from the proposed approach the average response time is very less and any kind of reliable request can be granted with in the shortest period of time. The maximum and the

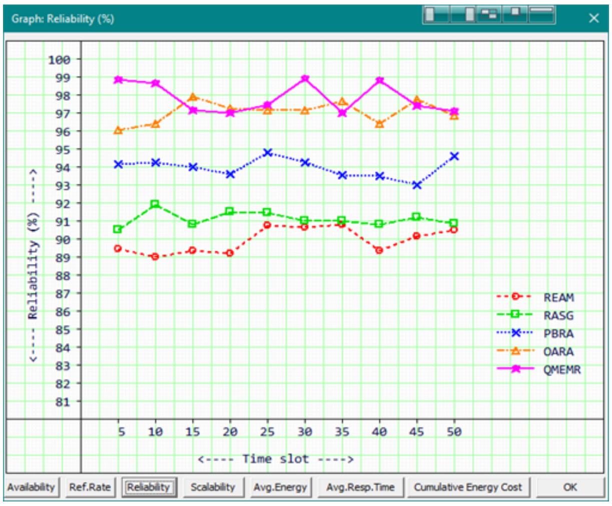

Figure 9: : Reliability graph for the existing and proposed method

Scalability (%)					
Time	**RAEM**	**RASG**	**PBRA**	**OARA**	**QMEMR**
5	90.24	89.16	92.89	96.72	99.24
10	90.95	88.72	92.14	96.63	98.7
15	90.69	88.42	92.21	96.69	97.89
20	90.03	89.07	91.42	97.73	99.23
25	90.73	88.17	92.99	96.39	99.31
30	90.28	89.85	92.48	97.59	98.57
35	90.51	89.37	92.82	97.95	98.28
40	89.88	89.29	91.99	97.71	98.77
45	89.09	89.48	92.29	97.15	99.48
50	89.44	89.17	91.19	97.92	99.4

Table 5: Scalability

It is denoted as Scalability $S = \frac{r_{consumed}}{r_{allocated}} > r_{threshold}$

Figure 10: : Scalability graph for the existing and proposed methods

Avg. Response Time (mS)					
Time	**RAEM**	**RASG**	**PBRA**	**OARA**	**QMEMR**
5	352	314	292	225	194
10	350	325	278	209	200
15	364	322	282	205	201
20	362	310	275	216	182
25	337	303	291	215	188
30	339	328	288	213	199
35	351	311	290	210	198
40	359	327	299	208	193
45	352	328	274	201	183
50	346	301	277	219	201

Table 6: Scalability

Figure 11: : Average Response time graph for the existing and proposed method

minimum average response time from proposed algorithm are 201ms, 183ms.

Cumulative energy cost: Individual spent cost for the consumption of energy for the resources in cloud environment. Lessor its value more will be the revenue or the profit.

Cumulative Energy Cost (uJ)					
Time	**RAEM**	**RASG**	**PBRA**	**OARA**	**QMEMR**
5	164	150	166	123	101
10	153	151	166	129	117
15	159	141	160	118	112
20	161	135	170	116	104
25	156	135	166	121	106
30	164	138	166	114	105
35	153	144	171	130	110
40	166	133	158	117	109
45	154	150	172	114	98
50	163	146	160	128	105

Figure 12: Cumulative Energy Cost

Cumulative energy cost is inversely proportional profit / revenue earned on the resource. From the above table it is clearly observed that out of present existing methods our proposed method is having less cumulative energy cost which indicates the more revenue is generated when energy cost distribution is followed by the proposed algorithm. The lowest value earned by the approach is 98 uJ, indicating more revenue at that point.

Figure 13: : Cumulative Energy cost graph for the proposed and existing methods

7 Conclusion

In this paper, we primarily focused on developing an effective algorithm that can be used to deliver QoS services while minimising energy consumption and maximising income. When the obtained values are compared to the state of the art of existing methods, our findings are found to be strong and accurate. There is also a need to achieve 100 percent reliability using innovative methods, and the total energy cost must be further reduced, as well as the energy consumption by resource. Other parameters can be considered in addition to these, depending on the domain in which the algorithm is being used.

Declaration of Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Conflict of interest

The authors declare no conflict of interest.

References

[1] R. Feldman, 2013. Techniques and applications for sentiment analysis, *Communications of the ACM, Vol.56*, No.4, Pp.82-89,

[2] M. Ghiassi, J. Skinner and D. Zimbra, 2013. Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network, *Expert Systems with applications, Vol.40*, No.16, Pp.6266-6282,

[3] N. Zainuddin, A. Selamat and R. Ibrahim, 2016. Improving twitter aspect-based sentiment analysis using hybrid approach, *Asian conference on intelligent information and database systems*, Pp.151-160,

[4] D. Antenucci, M. Cafarella, M. Levenstein, C. Ré and M.D. Shapiro, 2014. Using social media to measure labor market flows, *National Bureau of Economic Research*, Pp. 1-50,

[5] N. Zainuddin, A. Selamat and R. Ibrahim, 2018. Hybrid sentiment classification on twitter aspect-based sentiment analysis, *Applied Intelligence, Vol.48*, No.5, Pp.1218-1232,

[6] G. Wang, J. Sun, J. Ma, K. Xu and J. Gu, 2014. Sentiment classification: The contribution of ensemble learning, *Decision support systems, Vol.57*, Pp.77-93, 2014.

[7] O. Kolchyna, T.T. Souza, P. Treleaven and T. Aste, 2015. Twitter sentiment analysis: Lexicon method, machine learning method and their combination, *Computation and Language (cs.CL)*, Pp.1-32,

[8] N. Öztürk and S. Ayvaz, 2018. Sentiment Analysis on Twitter: A Text Mining Approach to the Syrian Refugee Crisis, *Telematics and Informatics, Vol.35*, No.1, Pp.136-147,

[9] K. Philander and Y. Zhong, 2016. Twitter sentiment analysis: Capturing sentiment from integrated resort tweets, *International Journal of Hospitality Management, Vol.55*, Pp.16-24.

[10] A. Hasan, S. Moin, A. Karim and S. Shamshir band, 2018. Machine learning-based sentiment analysis for twitter accounts, *Mathematical and Computational Applications, Vol.23*, No.1, Pp.1-15,

[11] A. Alnawas and N. Arici, 2019. Sentiment analysis of Iraqi Arabic dialect on Facebook based on distributed representations of documents, *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), Vol.18*, No.3, Pp.1-17,

[12] S.A. Salloum, C. Mhamdi, M. Al-Emran and K. Shaalan, 2017. Analysis and classification of Arabic newspapers' Facebook pages using text mining techniques, *International Journal of Information Technology and Language Studies, Vol.1*, No.2, Pp.8-17,

[13] K.M. Nahar, A. Jaradat, M.S. Atoum and F. Ibrahim, 2020. Sentiment analysis and classification of arabjordanian facebook comments for jordanian telecom companies using lexicon-based approach and machine learning, *Jordanian Journal of Computers and Information Technology (JJCIT), Vol.6*, No.03, Pp.247-263,

[14] H. Tran and M. Shcherbakov, 2016. Detection and prediction of users attitude based on real-time and batch sentiment analysis of facebook comments, *International conference on computational social networks*, Pp.273-284,

[15] T.H. Soliman, M.A. Elmasry, A. Hedar and M.M. Doss, 2014. Sentiment analysis of Arabic slang comments on facebook, *International Journal of Computers & Technology, Vol.12*, No.5, Pp.3470-3478,

[16] M. Meire, M. Ballings and D. Van den Poel, 2016. The added value of auxiliary data in sentiment analysis of Facebook posts, *Decision Support Systems, Vol.89*, Pp.98-112,

[17] A. Ortigosa, J.M. Martín and R.M. Carro, 2014. Sentiment analysis in Facebook and its application to e-learning, *Computers in human behavior, Vol.31*, Pp.527-541,

[18] F. Millstein, 2020. *Natural Language Processing With Python: Natural Language Processing Using NLTK*, Frank Millstein, Pp.1-116,

[19] A. Jabbar, S. Iqbal, A. Akhunzada and Q. Abbas, 2018. An improved Urdu stemming algorithm for text mining based on multi-step hybrid approach, *Journal of Experimental & Theoretical Artificial Intelligence, Vol.30*, No.5, Pp.703-723,

[20] M.A. Fauzi, 2018. Word2Vec model for sentiment analysis of product reviews in Indonesian language, *International Journal of Electrical and Computer Engineering, Vol.9*, No.1, Pp.525-530,

[21] A.I. Pratiwi, 2018. On the feature selection and classification based on information gain for document sentiment analysis, *Applied Computational Intelligence and Soft Computing*, Pp.1-5,

[22] S.V. Georgakopoulos, S.K. Tasoulis, A.G. Vrahatis and V.P. Plagianakos.2018., Convolutional neural networks for toxic comment classification, *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*, Pp.1-6,

[23] https://www.kaggle.com/mchirico/cheltenham-s-facebook-group

[24] Ragab et al., (2020). A Novel One-Dimensional CNN with Exponential Adaptive Gradients for Air Pollution Index Prediction, *Sustainability, 12*(23): 10090. https://doi.org/10.3390/su122310090

[25] Nasir et al. (2020). Pearson Correlation-Based Feature Selection for Document Classification Using Balanced Training, *Sensors, 20*(23): 6793, https://doi.org/10.3390/s20236793

[26] Huang, et al. (2019). Signal status recognition based on 1DCNN and its feature extraction mechanism analysis. *Sensors, 19*(9), https://doi.org/10.3390/s19092018

[27] Meliboev et al. (2020). CNN Based Network Intrusion Detection with Normalization on Imbalanced Data, *International Conference on Artificial Intelligence in Information and Communication*, 19-21 Feb. 2020, Japan. https://doi.org/10.1109/ICAIIC48513.2020.9064976

[28] Li, et al. (2020). A hybrid CNN-LSTM model for forecasting particulate matter (PM2.5), *IEEE Access, 8*, 26933-26940. https://doi.org/10.1109/ACCESS.2020.2971348

Cite this paper as:
Umamageswaran Jambulingam; K. Balasubadra K. (2022). A Unique Multi-Agent-Based Approach for Enhanced QoS Resource Allocation in Multi Cloud Environment while Maintaining Minimized Energy and Maximize Revenue, *International Journal of Computers Communications & Control*, 17(2), 4296, 2022.
https://doi.org/10.15837/ijccc.2022.2.4296