EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Integrative Resource Management in Multi Cloud Computing: A DRL Based Approach for multi-objective Optimization

Ramanpreet Kaur[1,2,*], Divya Anand[3], Upinder Kaur[4] and Sahil Verma[5]

[1]Department Of Computer Application, Lovely Professional University, Phagwara, Punjab, India
[2]Department of Computer Application, Baba Farid College of Engineering and Technology, Bathinda, Punjab, India
[3]Department of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab, India
[4]Department of Computer Science and Engineering, Akal University, Talwandi Sabo, Punjab, India
[5]Department of Computer Science and Engineering, CGC Jhanjeri, Mohali, India

Abstract

INTRODUCTION: The multi-data canter architecture is being investigated as a significant development in meeting the increasing demands of modern applications and services. The study provides a toolset for creating and managing virtual machines (VMs) and physical hosts (PMs) in a virtualized cloud environment, as well as for simulating various scenarios based on real-world cloud usage trends.
OBJECTIVES: To propose an optimized resource management model using the Enhanced Flower Pollination algorithm in a heterogeneous environment.
METHODS: The combination of Q-learning with flower pollination raises the bar in resource allocation and job scheduling. The combination of these advanced methodologies enables our solution to handle complicated and dynamic scheduling settings quickly, making it suited for a wide range of practical applications. The algorithm finds the most promising option by using Q-values to drive the pollination process, enhancing efficiency and efficacy in discovering optimal solutions. An extensive testing using simulation on various datasets simulating real-world scenarios consistently demonstrates the suggested method's higher performance.
RESULTS: In the end, the implementation is done on AWS clouds; the proposed methodology shows the excellent performance by improving energy efficiency, Co2 Reduction and cost having multi-cloud environment
CONCLUSION: The comprehensive results and evaluations of the proposed work demonstrate its effectiveness in achieving the desired goals. Through extensive experimentation on diverse datasets representing various real-world scenarios, the proposed work consistently outperforms existing state-of-the-art algorithms.

Keywords: Multi-cloud, Deep reinforcement learning, Resources allocation, Cyber shake seismogram workflow, Task scheduling Enhanced Flower Pollination

Received on 15 January 2024, accepted on 03 April 2024, published on 10 April 2024

doi: 10.4108/eetsis.5716

1. Introduction

Based on daily consumer need the demand for various services like SaaS, PaaS, and IaaS, promoting everyone to in the environment having multiple cloud. The concept of multi-cloud is used when the services are fulfilled from the various cloud, or the services are moved from one cloud to another cloud. The author presents taxonomy related to multi-cloud mentioned that the main actor that works in multi-cloud is Cloud Service Provider (CSP) [1]. Cloud computing provides high performance and a large number of services to the cloud user based on pay–per–usage. The cloud resource is distributed in various locations and connected according to a geographical area. These services are not allocated physically to the user but rather can be used on a rent basis. Cloud user works with multi-cloud

[*]Corresponding author. Email: **ramaninsa1990@gmail.com**

without knowing the physical location of the cloud which is fulfilling their request, same as virtual machines are also unknown to the user's location. Service Level Agreement (SLA)is an important part between cloud users and cloud providers because it deals with important parameters such as quality, pricing, security, etc. [2]. So, to fulfill the quality of services based on SLA, there is a requirement for efficient techniques of resources management. Various researchers are doing work in this field by providing various features of optimized resource allocation techniques.

Task scheduling is also very important with optimized resource allocation. Task scheduling means scheduling the incoming task in such a manner that efficient resources can be allocated. Various task scheduling approaches are doing well in this field. The suitable technique can be adopted based on the suitable requirement of cloud users or the resources allocation technique. Lots of work is done for resource allocation and task scheduling in cloud computing. But the same concept becomes very complicated in the case of a multi-cloud environment. As the services and users in multi-cloud are increased, the concept of task scheduling and resource allocation becomes a challenge for researchers. DRL technique of machine learning performs the best role in every IT-related field. It can make a very complex decision which was not possible in previous machine learning. In the cloud and multi-cloud, DRL also performs an important role in traffic, identification, future prediction, task scheduling, and resource allocation.

The performance of cloud or multi-cloud can be measured based on performance indicators or some parameters such as makespan, reliability, throughput, time, cost, power, and carbon emission. Deep reinforcement learn in intelligence elegant techniques are performing an amazing role to improve resource allocation in a multi-cloud environment without any SLA violation.

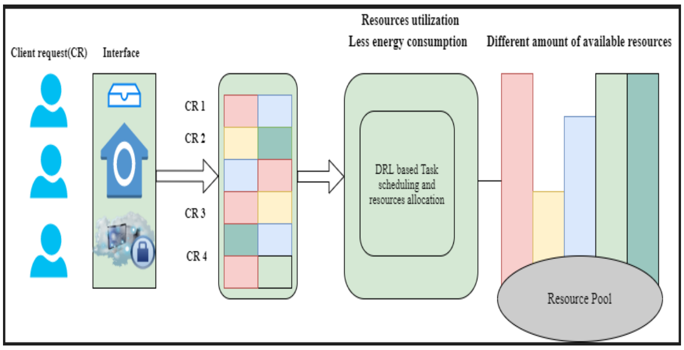


Figure 1. Representation of resources allocation in multi cloud environment.

Poor resource allocation or under-utilization of resources is responsible for high consumption of energy, cost, and some other precious parameters. High energy consumption leads to a high ratio of CO_2 and both deeply affects social life. The balance between both is a very challenging concept. The future of any country depends on healthy citizens. The deep reinforcement learning technique intelligently allocates resource to the user from more than one cloud to fulfil their requests having all benefits as energy efficiency, cost and CO_2reduction. The enhanced flower pollination algorithm (EFPA) works in the particular cloud for resource allocation based on local and global optimization techniques for energy efficiency. MET algorithm and cybershake seismogram workflow segregate the incoming requests and prepare a task queue for EFPA so that requests can be processed based on minimum execution time. This model is evaluated with cloud sim simulator. The remainder part of this paper includes as following. Section 2 is the related work and parameter-based study of task and resource allocation in the multi-cloud. Section 3 is the explanation of the proposed model having a flowchart and algorithms. The experiment results discussion and comparison of proposed work is in section 4.

2. Related Work

This section gives a basic review of various task scheduling and resource allocation techniques. Authors doing well to reduce various issues in the field of cloud and multi-cloud environments.

Predicting the future user's requirements in the cloud helps to reduce violations in service level agreements. But it is very difficult to predict the future requirement in the case of multi-cloud. Future prediction helps the cloud provider to allocate quality of service to the user. The author proposed a hybrid approach for future requirement prediction. This approach uses lazy learning, modified K-medoids, and lower bound dynamic time warping. This proposed approach gives better prediction when compared with others [3]. Resources management in multi-cloud needs a unique interface and wrapper for every service. The author proposed an approach that is adopted by deployable services in terms of open sources available platforms. This interface is different at run time, design, and deployment stages. The focus of this paper is to give an open-source, module-based solution that can be easily used [4]. The author presented how Cloud MF makes techniques of model-driven and ideas for minimizing the vendor lock-in and helps for allocations applications of multi-cloud. The Cloud ML (Cloud Modelling Language) permits to provision and deployment of applications in models of cloud provider-independent [5]. The author developed an optimized approach to minimize micro services repair, latency overhead of allocating containers on the cloud and reduce services cost. As micro-service are arranged in a container and that container will be allocated to VM but how to allocate the container on a suitable VM and allocate VM on a suitable cloud is a challenging issue. The author implements the NSGA-II genetic algorithm and compares it with the Greedy First-Fit algorithm; the implemented algorithm gives 300% improvement as compared to others [6].

Services provided by the cloud make every task very flexible as a business also moved toward the cloud and getting more benefits. To work on a single cloud is very efficient. But difficult in the case of multi or cross-cloud. To handle different instances of business processes near customers can be beneficial; the author presents a novel architecture of the environment of multi-cloud business provision. This architecture involves components to handle the monitoring and adaption of the business processes in a multi-cloud environment. This framework explains all about services such as Iaas, PaaS, monitoring, adoptions, etc. that can help to do business processes in a better way [7]. The author presented Replica aware task scheduling method to reduce the response delay of services. According to this algorithm, transferring computation and transfer data are combined. Resources matching is accomplished according to the availability of nodes. Failed or non-local data is replicated in advance to the targeted node. According to the cache placement algorithm, the next execute task is predicted. The experiment result shows that our proposed method performs better as compared to the benchmark algorithm in terms of node prediction and response time [8].

An integer linear programming model is developed to handle the scientific workflow in a multi-cloud environment. This helps to reduce financial costs by encountering the deadline requirement of the user. In this proposed model the resource limit imposed by the cloud provider and cost is calculated on an hourly basis. The experiment results show that change in deadline and workflow affect cost i.e., greater in CPU intensive workflows rather than other elasticity values remain always a constraint in work under long deadlines. A short deadline has a high cost. The comparison of the proposed model is done with the MIP-CG, MCPCPP, and IP-FC. The result shows that the proposed model is suitable for all deadlines; in the future, the makespan and total cost should be considered [9].

The author proposed Multiple-replica integrity auditing schemes for secure data storage on the cloud. A Cloud user is continuously taking data storage services on the cloud free from cost burden. The Author also mentioned open issues and research directions [10]. The author presents a hybrid formal verification approach for accessing high-quality service composition in the environment of multi-cloud by reducing the no. of the cloud provider. The proposed approach is helpful for checking the user request, services selection, and multi-cloud composition. Results show that this method reduces memory consumption [11]. This paper investigates resource management in a multi-cloud environment. The author also investigates the user's demand for applications in a multi-cloud environment. Definition and resources classification in the multi-cloud environment and three taxonomies of multi-cloud are mentioned. Future trends and challenges also point out [12]. In this paper, the author focuses on scheduling techniques that handle challenges in inter-cloud and also presents basic concerns and task scheduling related to multi-cloud. All scheduling techniques are categories based on some parameters. After containing the survey, the author mentions that security and load balancing is an important concern that should be considered in the future [13]. This author proposed a cloud-enabled workflow science gateway. This paper includes all the principles of integrating the cloud system with a science gateway. It integrates the WS-PGRADE/g USE and cloud broker platform. This integration is used in the CloudSME project where 20 companies port the simulation application on the cloud. The proposed method provides cloud access flexibility, and the user can access all clouds integrated with this gateway [14]. During the use of multi-cloud, the user has to face some challenges such as provisioning, elasticity, portability, and availability. To handle these challenges the author presented the so cloud framework that is deployed on 10 cloud providers' complete architecture and interaction between all components of the so Cloud framework discussed in this paper. With this approach, the user can get high availability [15].

The author focuses on the problem of VM placement for reducing cost and saving energy in a heterogeneous environment of multi-cloud. The author proposed a mimetic algorithm for VM placement based on cost-efficient to solve this problem i.e .called grouping genetic. The proposed algorithm reduces running PM and consumption of energy by the geographical distribution of the data center. Hill climbing is also used do searching in local to maximize the speed and run time of the genetic algorithm. Comparison of the proposed model is made with three other recent research and found that the proposed model performs better to reduce cost and energy [16]. In this paper, the author developed normalized hybrid service brokering With Throttled Round Robin Load Balancing (NHSB_TRB) to provide cost-effective services to the user. This approach produces a normalized value of optimized cost. The data center based on cost is selected for distributing the load. The weighted threshold is used to distribute the load on the data center and the round-robin load balancing approach distributes the load on VM. The experiment result shows that the proposed model improves response time, monetary cost, and processing time of data center up to 17.39%, 7.06%, and 31.35 when compare with ORT_RR, CDC_RR, ORT_THR, and ORT_ES approaches [17]. Table 1 compares task scheduling and resources allocation techniques in a multi-cloud environment based on a parameter such as Makespan, cloud/resource, utilization, scalability, cost, Response Time, Energy efficiency, and Co2 reduction.

Table 1. Task scheduling and resource allocation techniques in a multi-cloud environment.

Ref. No	Tecnique	Makespan	resources utilization	Scalability	Cost	Response Time	Energy efficiency	Co_2 Reduction
SK Panda [18]	AMinB, AMaxB, and AMinMaxB task scheduling algorithm	✔	✔	×	×	×	×	×
Lijin P [19]	Game theory-	✔	×	×	×	×	×	×

	based resources allocation algorithm							
A. Pietrabissa [20]	Q-learning(Policy reduction and state aggregation strategy)	×	×	✔	×	×	×	×
SK Mishra [21]	Energy-Aware Task Allocation in Multi-Cloud Network.	×	×	×	×	×	✔	×
J.Carvalho [22]	Simple Adaptive weighting method and multi-choice knapsack problem	×	✔	×	✔	✔	×	×
P.Antonio1[23]	SARSA(λ) and Q-learning	×	×	×	×	×	×	×
S.Kang [24]	DSS	×	×	×	×	×	×	×
Z.Chen [25]	OWS-A2C	×	✔	×	✔	×	×	×
SK Panda[26]	MCC,MEMAX,CMMN Algorithm	✔	✔	×	×	×	×	×
M.Farid [27]	FR-MOS	✔	✔	×	✔	×	×	×
T.Subramanian [28]	Novel cloud brokering architecture	×	×	✔	×	×	×	×
C. Thirumalaiselvan [29]	ELB, high priority scheduling algorithm, and rate-based scheduling algorithm.	✔	×	×	×	✔	✔	×
N.Grozev[30]	Rule-based domain-specific model.	×	×	×	×	×	×	×

The above table shows that not a single technique works collectively on cost, energy, and Co$_2$. so, the same should be considered in research work.

3. Proposed methodology

With the time-varying demands of construction, optimization, and user requests, a Cloud computing system cannot be considered self-sufficient. The internal environment for agent state and decision-making are also temporal shifting like Cloud computing's income ratio, which varies in different periods during a single day [31]. A DRL framework with time-varying external stimuli is used to view the decision-making process as an ensemble. Each mapper's information is as follows: Extrinsic or internal stimuli are inputs to the mapper of time-varying, whereas the output is a change in the agent's state or state-changing. It is a mapper of stimulus evolution in agent and state, where the stimulus force is input and the set of agent-state at real-time is output, as the agent and state are usually changing with stimuli.

When it comes to managing resources in a cloud computing environment, resource allocation (RA) is one of the methods available. When it comes to establishing the optimal balance between VM and PMin the cloud data center, Infrastructure as Service providers confronts a significant difficulty. This is known as finding the optimal allocation for VMs and PMs in terms of the number of resources they require [32]. There are two components to resolving the issue of VM allocation: first, the acceptance of new requests for VM provisioning and the placement of the VMs on a PM, and second, the optimization of existing VMs.

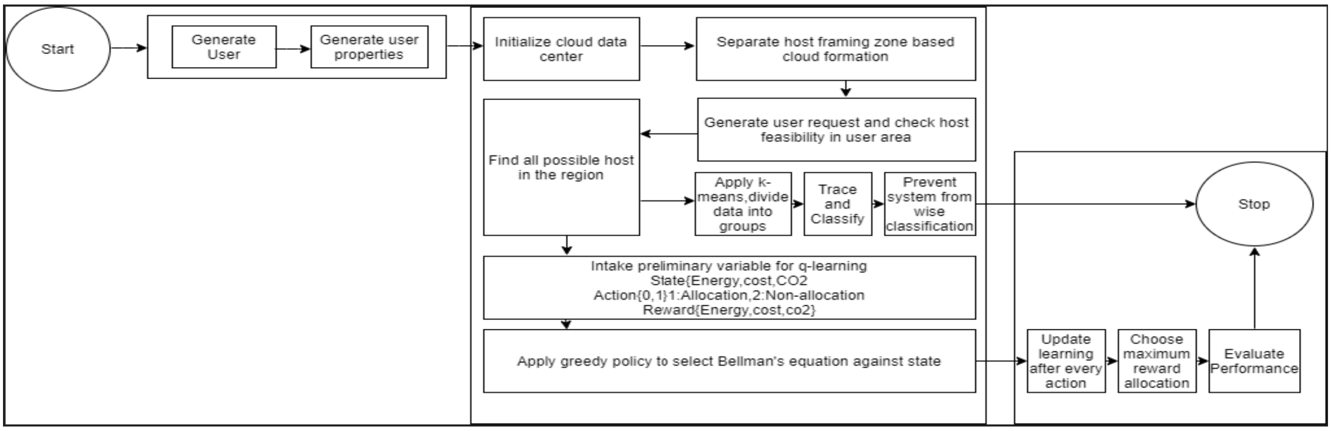

Start
Generate User
Generate user properties
Initialize cloud data center
Separate host framing zone based cloud formation
Generate user request and check host feasibility in user area
Find all possible host in the region
Apply k-means,divide data into groups
Trace and Classify
Prevent system from wise classification
Stop
Intake preliminary variable for q-learning
State{Energy,cost,CO2
Action{0,1}1:Allocation,2:Non-allocation
Reward{Energy,cost,co2}
Apply greedy policy to select Bellman's equation against state
Update learning after every action
Choose maximum reward allocation
Evaluate Performance

Figure 2. Workflow of DRL in a cloud environment

The decision mapper is in charge of computing the next action based on the agent's present state, and the action taken at the next opportunity is the output. To put it in layman's terms: The environment is fed by actions provided by a mapper and grows as a result of those actions. Environment's output is fed into mapper of feedback, while agent's internal stimuli are fed into mapper of time-varying [33-35]. Replay storage is used to store long-term input in preparation for future use, while timely feedback is taken from the environment. The settings of the decision-maker will be updated as a result of both long-term and real-time feedback. Generalized RL built on the integration of mappers. Programs are often used to model time-varying, stimulus evolution, and environmental conditions in some research. Neural networks can be used to build a decision and feedback mapper. The feedback mapper can be implemented as a neural network to calculate the loss function of the neural network in the decision mapper since it aims to update the parameters of the decision-maker. In computing, a VM (virtual machine) is an emulation of a certain computing system that is used to simulate another system. Virtual machines are capable of running since they are based on the computer architecture and functionalities of a real or physical computer. These systems can be implemented using specialized hardware and software, or they can be implemented using a combination of both [36]. Virtual machines can be divided into several categories based on how closely they resemble their real-world counterparts in terms of capability.

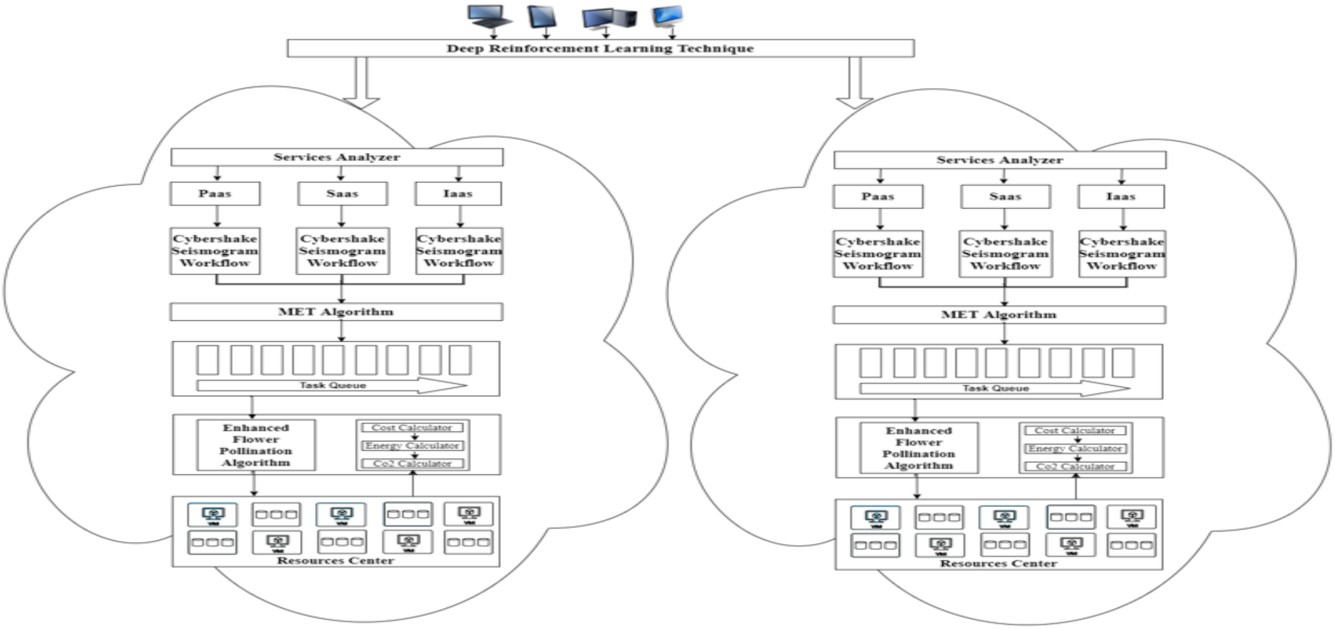


Figure 3. Multi-objective resource optimization deep reinforcement learning (MOROT-DRL) model.

As a result, system virtual machines can serve as a complete substitute for the targeted virtual machines, as well as providing the amount of functionality required to operate an operating system (also known as full virtualization VMs). While this is true, a process virtual machine provides an abstracted and platform-independent execution environment for a single computer application running on a variety of platforms. The centralized cloud resource manager is in charge of managing the resources in cloud computing. Cloud data center (DC) resources are made available to cloud consumers using virtual machines (VMs), which are based on physical machines (PMs). To maximize resource use while minimizing energy consumption, cloud computing Infrastructure as a Service (IaaS) providers must implement dynamic resource management techniques in their cloud DCs. Because of business considerations, the resource management strategies and algorithms used in public clouds are not revealed. The proposed energy-efficient resource allocation mechanism is comprised of a single central scheduling point (CSP) and N cloud users. CSP managed many heterogeneous resources like memory, processing units, network bandwidth, and so on in the form of virtual machines (VMs). When these virtual machines (VMs) were requested by cloud customers to complete their activities, the cloud provider allocated them based on the preferences of the cloud consumers. The primary purpose of this proposed Enhanced flower pollination algorithm is to reduce the amount of energy consumed during work scheduling in the cloud environment, as well as to reduce the number of task scheduling issues in cloud computing.

```
Algorithm 1: Enhanced flower pollination algorithms
1: Start
2: Input
Data center structure $D_s$
Size of Population $S_m$
Total number of takes $S_{m,\ task}$
Number of iterations $i_{max}$
Number of Virtual machines $S_{m,\ vm}$
3: Output
        Optimal Solution $Y_a$
4: Calculate the global task queue
$Y_q^{j+1}=Y_q^j+\Upsilon(\Psi)(S-Y_q^j)$
5: Calculate the local task queue
$Y_q^{j+1}=Y_q^j+ɯ(Y_i^j -Y_m^j)$
6: finally find out the Optimal Solution
        $Y_a$, a=1,2,3,………$S_m$
                i = i+1;
7: Update and  repeat the Optimal Solution
8: Select the best solution
9: Stop
```

Increased energy use results in an increased operating expense. The most pressing issue is the increase in excessive carbon emissions (CO2). It has a greater impact on the environment. This limited supply must be put to good use. The most critical step is to reduce the use of energy and power. It is important to avoid wasting resources. This is referred to as energy conservation. The efficient use of resources can be improved by utilizing virtualization technologies. The dynamic consolidation of virtual machines (VMs) is made possible by virtualization technology. Cloud service providers can host several virtual machines (VMs) on a single physical server, i.e. , virtualization. One technique to reduce power usage is to turn off nodes that are not in use.

E_{CE} is a metric that measures how much energy is expended while a job is in the process of being set up for execution, such as copying the data needed for the task to run. The amount of CPU energy used to carry out the task in the designated VM is therefore considered as an E_{PE}, which is directly related to the amount of CPU energy utilized. Thus, the total amount of energy consumed by the user jobs in the Task Set may be calculated using Equation 1.

$$E_{Total} = \sum E_{VM} \quad (1)$$

Here E_{VM} is known as $E_{VM} = E_{CE} + E_{PE}$ (2)

Algorithm 2: Task Scheduling Algorithm for Deep reinforcement learning

```
1: Start
2: evaluate the resources information of task
3:  Set $E_{VM}$ task.
4: Evaluate$E_{total}$
        $E_{Total} = \sum E_{VM}$
        $E_{VM} = E_{CE} + E_{PE}$
5: Do till all task mapped
6: Earliest completion time and resources of all task are calculated
7: Set resource's ready time
8:  According completion time set all resources
9: Do for all R
10: calculate highest Completion Time of  Ti
If Maximum Completion Time <makespan
Compute makespan = max(CT(R))
11: Figure out task having minimum completion time.
12: Find out $T_i$ with minimum ET
13: Reschedule task $T_i$ according to produced resources.
14: Change ready time for those resources
15: End
```

4. Results and Discussion

4.1 Application Initialization at AWS cloud

The proposed work stands as a testament to the adaptability and scalability of our approach. Not content with mere theoretical considerations, our endeavor has ventured into the realm of real-time application on the AWS (Amazon Web Services) cloud platform, utilizing multiple cloud instances to amplify its effectiveness.

At the heart of this initiative lies the creation of a comprehensive virtual server, constructed with a robust Java-based architecture. To ensure the smooth operation of our endeavor, we meticulously installed the requisite Java Development Kits (JDKs) to empower our virtual server with the necessary tools and capabilities. What sets our approach apart is the deliberate segmentation of responsibilities between two distinct cloud instances, each strategically situated in separate cloud data centers.

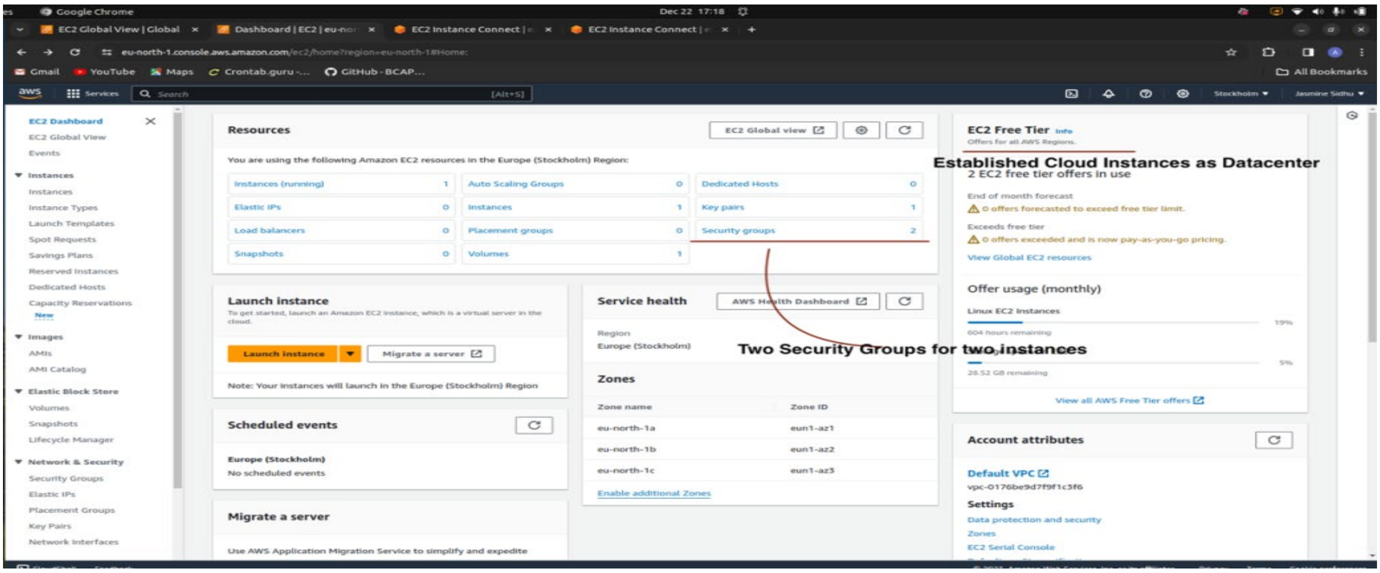
One of these instances is entrusted with the critical task of executing the proposed work, harnessing the computational power and resources offered by AWS to carry out complex operations with efficiency and speed. The other instance plays a complementary role, focusing on the crucial aspect of storage. Its primary responsibility is to manage and maintain the data and results generated during the execution phase, ensuring that valuable information remains organized and readily accessible.

4.2 Execution of cloud server

Executing the current file unfolds as a meticulously orchestrated process, with each step designed to unveil critical insights into the program's performance and its ecological consequences. The installation of indispensable libraries drawn from Cloud Sim and Apache POI binaries establishes the foundation for the subsequent execution. These libraries equip the program with the necessary tools and capabilities to operate effectively within a cloud computing environment.

The AWS instance and the proposed Cloud application have been seamlessly interconnected, forming a powerful synergy for executing various tasks and jobs within the AWS cloud environment. This integration enables the proposed Cloud application to leverage the computational capabilities and resources offered by AWS, ensuring efficient and reliable execution of its processes.

When tasks are initiated within the proposed Cloud application, the AWS cloud platform immediately springs into action. It dynamically allocates and manages the necessary computing resources and hosts to execute these tasks effectively. This on-demand resource provisioning ensures that the application can efficiently scale to accommodate varying workloads and demands, making it an ideal choice for handling diverse job requirements.

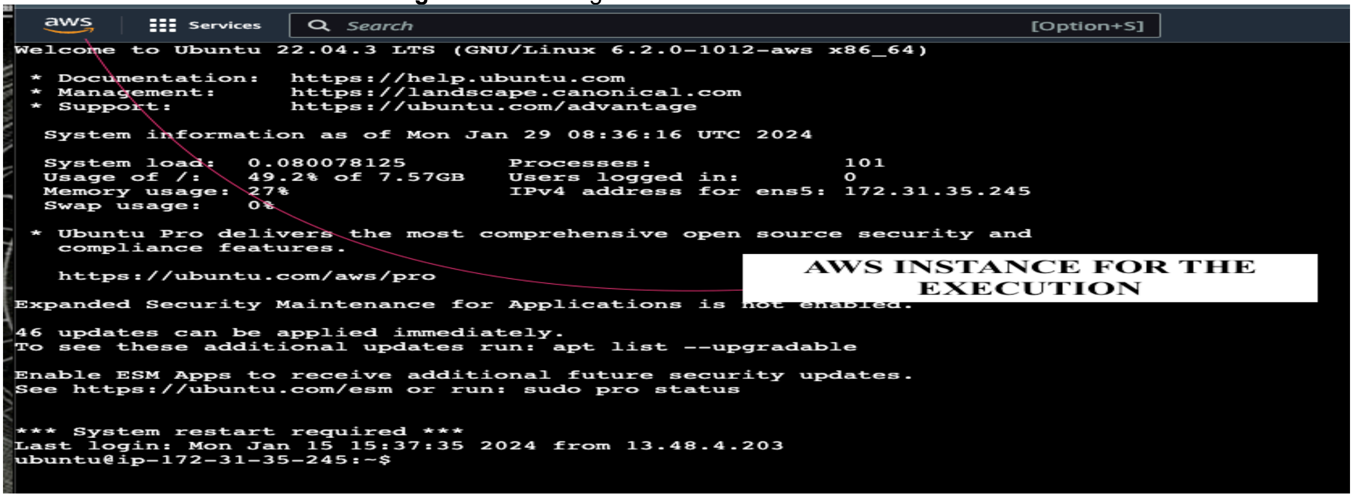


Resources

You are using the following Amazon EC2 resources in the Europe (Stockholm) Region:

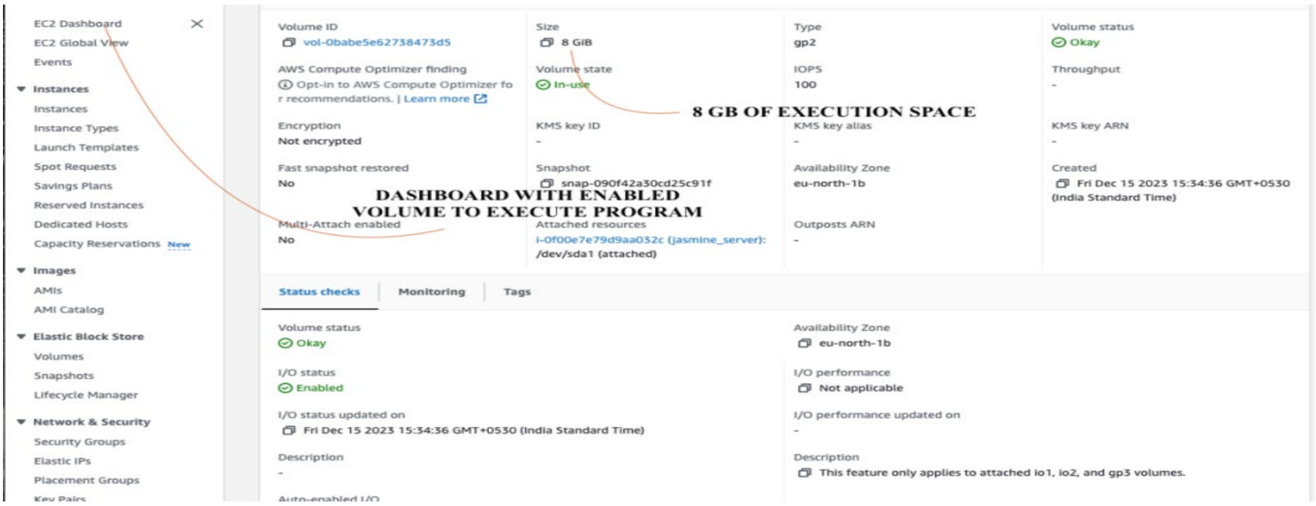
Instances (running)	1	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	1	Key pairs	1
Load balancers	0	Placement groups	0	Security groups	2
Snapshots	0	Volumes	1		

Figure 4.1 Running Instances on AWS Multi-cloud



```
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Jan 29 08:36:16 UTC 2024

  System load:  0.080078125       Processes:             101
  Usage of /:   49.2% of 7.57GB   Users logged in:       0
  Memory usage: 27%               IPv4 address for ens5: 172.31.35.245
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

46 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


*** System restart required ***
Last login: Mon Jan 15 15:37:35 2024 from 13.48.4.203
ubuntu@ip-172-31-35-245:~$
```

Figure 4.2 AWS Instance Creation



Volume ID vol-0babe5e62738473d5	Size 8 GiB	Type gp2	Volume status Okay
AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more	Volume state In-use	IOPS 100	Throughput -
Encryption Not encrypted	KMS key ID -	KMS key alias -	KMS key ARN -
Fast snapshot restored No	Snapshot snap-090f42a30cd25c91f	Availability Zone eu-north-1b	Created Fri Dec 15 2023 15:34:36 GMT+0530 (India Standard Time)
Multi-Attach enabled No	Attached resources i-0f00e7e79d9aa032c (jasmine_server): /dev/sda1 (attached)	Outposts ARN -	

Volume status Okay	Availability Zone eu-north-1b
I/O status Enabled	I/O performance Not applicable
I/O status updated on Fri Dec 15 2023 15:34:36 GMT+0530 (India Standard Time)	I/O performance updated on -
Description -	Description This feature only applies to attached io1, io2, and gp3 volumes.
Auto-enabled I/O	

8 GB OF EXECUTION SPACE

DASHBOARD WITH ENABLED VOLUME TO EXECUTE PROGRAM

Figure 4.3 AWS Dashboard with application and space

Once the AWS instance is allocated for the RamanCloud application, a dedicated storage space is created to cater to the user's data storage needs. In this particular setup, a total of 8 gigabytes (GB) of storage space is provisioned for the user.

This storage space plays a critical role in accommodating various data assets, files, and resources required by the user within the AWS environment. It ensures that the user has ample capacity to store, access, and manage their data efficiently as they interact with the RamanCloud application.

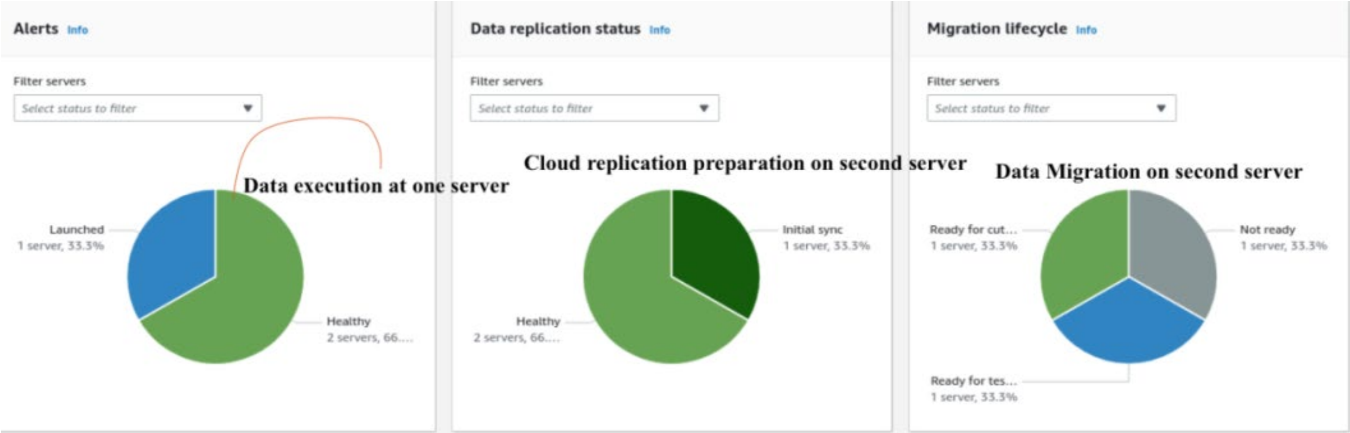
The 8 GB of allocated storage is designed to support a wide range of use cases, such as storing datasets, reports, configurations, and any other data pertinent to the user's activities within the application. This provisioned space not only facilitates data retention but also aids in ensuring smooth and uninterrupted operation of the RamanCloud application by allowing for the efficient organization and retrieval of essential information.

To ensure the program's execution aligns with specific objectives, instruction sets are thoughtfully crafted and passed. These sets serve as the program's guiding principles, dictating its tasks, workloads, and other intricate parameters that shape its behavior.

User and other virtual machine (VM) instances are then artfully configured within the chosen cloud environments. The allocation of computing resources—ranging from CPU capacity to memory and storage—aims to cater precisely to the program's unique requirements. This careful provisioning ensures optimal performance and resource utilization.

The true essence of the experiment comes to life when the program is executed simultaneously in two separate cloud environments, each hosting its own dedicated set of VM instances. This parallel execution serves as a crucible for comparative analysis, allowing for the identification of performance variations and resource utilization disparities between the two cloud setups.

In this setup, one server is dedicated to executing programs, while another server is solely responsible for storing data. These two servers operate independently, each with its specific role, without any direct sharing of computational tasks or data transfer between them.



Data execution at one server

Cloud replication preparation on second server

Data Migration on second server

Figure 4.4 Data Migration between multiple cloud

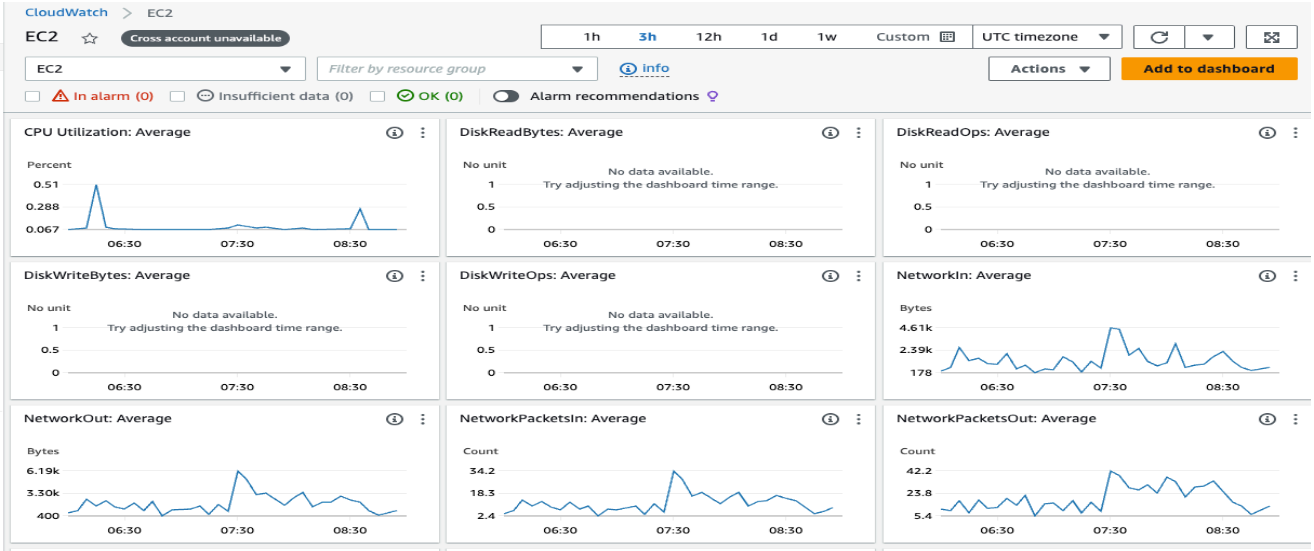


Figure 4.5 Execution Outcome at server 1 based on load

Simultaneously, AWS server 1 continuously monitors its CPU utilization. This real-time tracking provides valuable insights into how intensively the server's central processing unit is being used by the RamanCloud application. When the CPU utilization approaches or exceeds predefined thresholds, the application can make informed decisions to maintain system stability and performance.

For instance, when CPU utilization is high due to a surge in workload, the application might employ load balancing techniques to distribute tasks across multiple servers or allocate additional resources to AWS server 1. This ensures that the application can continue processing tasks efficiently without causing performance degradation or downtime. Conversely, during periods of low CPU utilization, resources can be allocated more sparingly to reduce operational costs, making the resource utilization process highly dynamic and responsive.

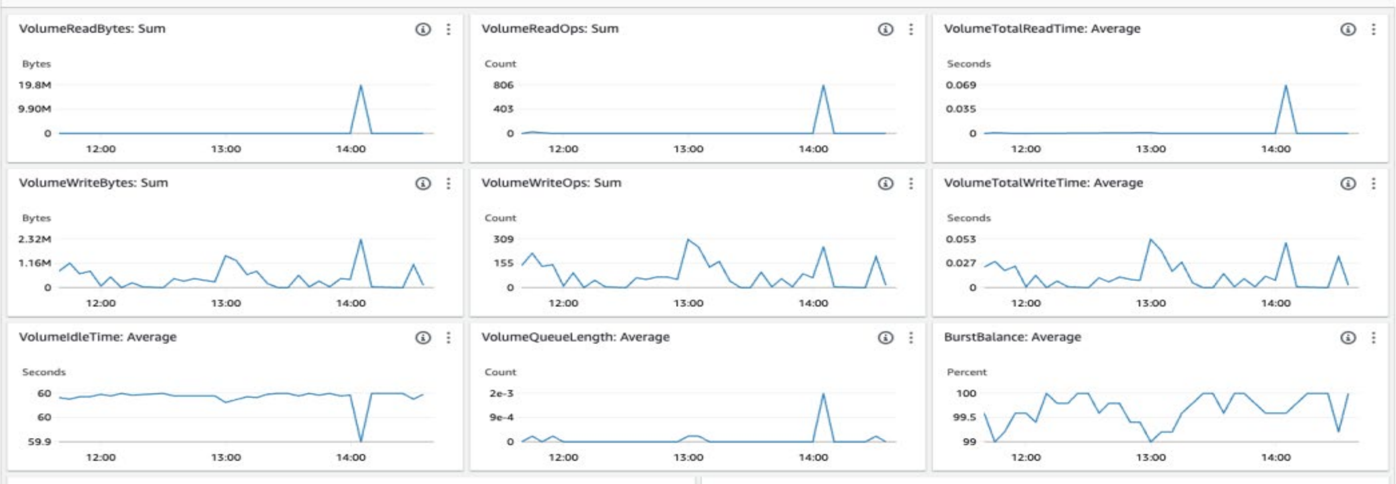


Figure 4.6 Read and Write Operations on server 2 for data storage.

The interplay between supplied load and CPU utilization metrics within AWS server 1 allows the proposed cloud application to operate efficiently in a scalable manner. It makes it possible for the application to adjust to shifting workloads and resource requirements in an efficient manner, guaranteeing peak performance and resource use all the time. The proposed cloud application's overall dependability and efficiency in the AWS environment are greatly enhanced by this dynamic monitoring and adjusting procedure.

The proposed work is expected to yield significant benefits, especially when lever-aging a real-time AWS cloud environment consistently. This choice offers unparalleled convenience, even for relatively small-scale projects, as AWS's real-time cloud infrastructure provides a noteworthy performance boost. This enhancement is primarily attributed to AWS's shared memory concept, which accelerates processing speeds.

In practice, utilizing AWS's real-time cloud platform can result in a remarkable 8% increase in processing speed compared to alternative cloud solutions. This boost in performance stems from the efficiency gained through shared memory resources, allowing for faster data access and computation. Consequently, even for tasks of limited scope, the AWS real-time cloud proves to be a valuable asset, optimizing overall processing times and ensuring that applications run smoothly and swiftly.

Analysis and comparison are where the experiment's true insights surface. Discrepan-cies in power efficiency, environmental sustainability, and program performance are scrutinized, potentially leading to actionable optimization strategies to enhance efficiency or reduce environmental impact.

The findings and outcomes are not left in isolation. Instead, they are meticulously documented and woven into comprehensive reports, ensuring the dissemination of valuable insights and facilitating data-driven decision-making for future cloud re-source allocation and environmentally conscious application deployment.

5. Conclusion

Multi-cloud gives more elasticity to the users by combining multiple cloud domains and data centers. These features attract not only normal users but also the biggest companies and businesses. The requirement of cloud providers and cloud users are escalating gradually. The challenge is to handle the request and allocate the required resource. Researchers proposed many scheduling and resource allocation techniques which give good results in various parameters such as time, cost, throughput, reliability, etc. Some more are needed to improve. So, for this, we proposed the MOROT-DRL model implemented in real cloud environment which works on energy, cost, and CO2 parameters. The Q-learning technique logically handles the incoming request and works as an intelligent model in a multi-cloud environment. Cyber shake seismogram workflow and minimum execution time algorithms create a queue based on minimum execution time and schedule the task in a specific cloud. The bio-inspired algorithm, i.e., enhanced flower pollination picks the task from the queue and allots the optimized resources with dynamic switching property, and local and global strategy. In the end, the implementation is done on AWS clouds, the proposed methodology shows the excellent performance by including multi-cloud environment.

References

[1] Petcu,D.: Consuming Resources and Services from Multiple Clouds,.J Grid Computing ,12(2) ,321–345(2014).

[2] Panda,S.K.,Jana,P.K.: SLA-based task scheduling algorithms for the heterogeneous multi-cloud environment. J Supercomputing,73(6), 2730–2762(2017).

[3] Keshavarzi, A., Haghighat, A.T., Bohlouli, M.: Enhanced time-aware QoS prediction in multi-cloud: a hybrid k-medoids and lazy learning approach(QoPC). Computing 102(4),923–949(2020).

[4] Munteanu,V.,Sandru,C.,Petcu,D.: Multi-cloud resource management: cloud service interfacing, Munteanuetal. Journal of Cloud Computing: Advances, Systems and Applications (2014).

[5] Ferry,N., Chauvel,F., Song,H., Rossini,A.: Lushpenko,M.,Solberg.:CloudMF: Model-Driven Management of Multi-Cloud Applications. ACM Transactions on Internet Technology18(2)(2018).

[6] Guerrero,C., Lera,I.,Juiz, C.:Resource optimization of container orchestration:a case study in multi-cloud microservices-based application J Supercomput74(7),2956-2983(2018).

[7] Kritikos,k.,Zeginis,C.,Iranzo,J.,Gonzalez,R.,Seybold,D.,Griesinger,F., Domaschk,J.:Multi-cloud provisioning of business processes.Journal of Cloud Computing: Advances, Systems and Applications(2019).

[8] Li,C.,Zhang,J.,Tang,H.: Replica-aware task scheduling and load-balanced cache placement for delay reduction in the multi-cloud environment.The Journal of Supercomputing, 75(5),2805-2836(2019).

[9] Mohammadi,S.,Pedram,H.,Karimi,.: Integer linear programming-based cost optimization for scheduling scientific workflows in multi-cloud environments.The Journal of Supercomputing 4717–4745(2018).

[10] Li,J.,Lin,Y.,Jia,X.,Ren,K.:Multiple-replica integrity auditing schemes for cloud data storage.Concurrency Computat Pract Exper(2019).

[11] Souri,A.,Rahmani,A.,Rezaei,N.:A hybrid formal verification approach for QoS-aware multi-cloud service composition. Cluster Computing(2453–2470)2020.

[12] Carvalho,J., Trinta,F., Vieira,D., Cortes,O.:Evolutionary solutions for resources management in multiple clouds: State-of-the-art and future directions. Future Generation Computer Systems, 88,284-296, 2018.

[13] Masdari,M.,Zangakani,M.: Efficient task and workflow scheduling in inter-cloud environments: challenges and opportunities.The Journal of Supercomputing,499-535(2019).

[14] Bruno,R.:Costa,F.,P,Ferreira.:free Cycles - Efficient Multi-Cloud Computing Platform", J Grid Computing,15(1)501–526(2017).

[15] Paraiso,F.,Merle,P.,Seinturier,L.:so Cloud: a service-oriented component-based PaaS for managing portability, provisioning, elasticity, and high availability across multiple clouds.Computing.539–565(2016).

[16] Rashida,S.,Sabaei, M.,Ebadzadeh, M.,Rahmani,A.:A memetic grouping genetic algorithm for cost-efficient VM placement in the multi-cloud environment,Cluster Computing 23(2),797–836(2020).

[17] Khan, M.:Optimized hybrid service brokering for multi-cloud architectures", The Journal of Supercomputing, 76,666–687(2020).

[18] Panda,S.,Gupta,I., Jana,P.:Task scheduling algorithms for multi-cloud systems: allocation-aware approa21, 241–259 (2019).

[19] Lijin P.: Resource Allohation in Multi-Cloud Based on Usage Logs.International Journal of Scientific Research in Computer Science, Engineering and Information Technology IJSRCSEIT 3(2018).

[20] Pietrabissa,A., Priscoli,F., Giorgio,A., Giuseppi,A.,Panfili,M., Suraci,V.:An Approximate Dynamic Programming Approach to Resource Management in Multi-Cloud Scenarios,International Journal of Control, 492-503(2016).

[21] Mishra,S., Mishra,S.,Alsayat,A.,Jhanjhi,N.,Humayun,M., Sahoo,K.,Luhach,A.:Energy-Aware Task Allocation for Multi-Cloud Networks, IEEE Access,8, (2020).

[22] Carvalho,J., Vieira,D., Trinta,F.:Dynamic Selecting Approach for Multi-cloud Providers", Springer International Publishing AG, part of Springer Nature , 37–51,(2018).

[23] Antonio,P., Stefano,B.,Francisco,F.: Alessandro,G., Guido,O., Martina ,P.,Vincenzo,S.:Resource Management in Multi-Cloud Scenarios via Reinforcement Learning.Proceedings of the 34th Chinese Control Conference July 28-30(2015).

[24] Kang,S., Veeravalli,B.,Aung,K.:Dynamic scheduling strategy with efficient node availability prediction for handling divisible loads in multi-cloud systems.J. Parallel Distrib. Computing, 113,pages 1-16(2018).

[25] Chen,Z., Lin,K., Lin ,B., Chen ,X., Zheng,X., Rong,C.:Adaptive Resource Allocation and Consolidation for Scientific Workflow Scheduling in Multi-Cloud Environments.IEEE Access, 8, 2020.

[26] Panda,S., Jana,P.:Efficient task scheduling algorithms for the heterogeneous multi-cloud environment. J Supercomput (2015) .

[27] Farid ,M.,Latip ,R., Hussin ,M.,Hamid,N.:Scheduling Scientific Workflow Using Multi-Objective Algorithm With Fuzzy Resource Utilization in Multi-Cloud Environment.IEEE Access ,8(2020).

[28] Subramanian,T.,Savarimuthu1,N.:Application-based brokering algorithm for optimal resource provisioning in multiple heterogeneous clouds. Vietnam J ComputSci,3(1) 57-70 (2016).

[29] Thirumalaiselvan,C., Venkatachalam,V.: A strategic performance of virtual task scheduling in multi cloud environment.Cluster Comput (2019).

[30] Grozev,N.,Buyya,R.:Regulations and latency-aware load distribution of web applications in Multi-Clouds. J Supercomput (2016).

[31] Zhan,W.,Luo,C. Wang,J., Wang,C., Min,G., Duan,H., Zhu,Q.:Deep-Reinforcement-Learning-Based Offloading Scheduling for Vehicular Edge Computing.IEEE Internet of Things Journal, 7(6), 5449-5465(2020).

[32] Qi,Qi.,Wang,J., Ma,Z., Sun,H., Cao,Y., Zhang,L.,Liao,J.:Knowledge-Driven Service Offloading Decision for Vehicular Edge Computing.A Deep Reinforcement Learning Approach.IEEE Transactions on Vehicular Technology, 68(5),4192-4203(2019).

[33] Wang,Y.,Liu,H., Zheng,W., Xia ,Y., Li,Y., Chen,P., Guo,K.,Xie,H.:Multi-Objective Workflow Scheduling With Deep-Q-Network-Based Multi-Agent Reinforcement Learning.IEEE Access, 7,39974-39982(2019.

[34] Baer.S.,Bakakeu,J.,Meyes,R.,Meisen,T.:Multi-Agent Reinforcement Learning for Job Shop Scheduling in Flexible Manufacturing Systems.2019 Second International Conference on Artificial Intelligence for Industries ,22-25(2019)

[35] Zhang,L., Wang,Q., Sun,H., Liao,J.:Multi-task Deep Reinforcement Learning for Scalable Parallel Task Scheduling.2019 IEEE International Conference on Big Data (Big Data), 2992-3001(2019).

[36] Shetty,C.; Sarojadevi,H.,Prabhu,S.:Machine Learning Approach to Select Optimal Task Scheduling Algorithm in Cloud.Turkish Journal of Computer and Mathematics Education, 12(6),2565-2580(2021).