AUT Jouranl of Modeling and Simulation

AUT J. Model. Simul., 52(1) (2020) 129-142
DOI: 10.22060/miscj.2020.17354.5180

Energy and Security Awareness Task Scheduling based on Fuzzy System in Cloud Computing

Najme Mansouri[1*], Behnam Mohammad Hasani Zade[2], Mohammad Masoud Javidi[3]

[1] Assistant Professor in Computer Science Department, Shahid Bahonar University of Kerman, Kerman, Iran
[2] MSc, Computer Science Department, Shahid Bahonar University of Kerman, Kerman, Iran
[3] Associate Professor in Computer Science Department, Shahid Bahonar University of Kerman, Kerman, Iran

ABSTRACT: The increasing popularity of cloud computing environments makes task scheduling as a critical problem and a hot research topic. It is necessary to decrease the energy related costs and enhance the lifespan of high performance computing resources used in cloud data centers. Moreover, the high quality of security service is increasingly critical for security-sensitive applications that work with large-scale data files such as bioinformatics. We propose a new task scheduling algorithm that includes: 1) analyzing task execution time based on the load of data centers; 2) modeling the resource utilization; 3) calculating security cost based on the failure probabilities; 4) evaluating power consumption based on the linear model; and 5) analyzing the closeness centrality of data centers to improve data retrieval time. Finally, it designs a fuzzy inference system with five inputs (i.e., total execution cost, resource utilization cost, security cost, energy consumption, and centrality) in order to assign a merit value to each data center for task execution. Cloud is a dynamic environment and there is not accurate information at every moment. Therefore, fuzzy inference is a good choice for predicting the behavior of the system and scheduling decisions. The simulation results indicate that the proposed algorithm obtains superior performances respectively in waiting time, success rate, energy consumption, and degree of imbalance around 14%, 12%, 15%, 11% on average than other similar methods under high load condition. Consequently, the proposed strategy has potentials to enhance the performance of QoS delivery since it can effectively utilize cloud resources.

Review History:

Received: Nov. 11, 2019
Revised: Jan. 16, 2020
Accepted: Feb. 16, 2020
Available Online: Jun. 15, 2020

Keywords:

Cloud computing
Task scheduling
Security
Energy consumption
Fuzzy system
Simulation

1- INTRODUCTION

Cloud computing is a computing standard, which consists of distributed resources to deliver on demand services and infrastructure through a network [1-2]. Cloud computing environment is the most popular computational system for unique characteristics that are shown in Fig.1. There are three cloud models as follows. 1) Infrastructure as a Service (IaaS) that provides infrastructure or virtual machines on demand. 2) Platform as a Service (PaaS) that is applied for applications and other developments. 3) Software as a Service (SaaS) uses the web to deliver applications that are managed by a third-party vendor [3-4].

The second classification of cloud is according to the deploying model. In a public cloud, infrastructure services (i.e., servers, storage, networks, development platforms etc.) are presented for public. Nevertheless, private cloud presents cloud resources for exclusive use by single organization. In community cloud, the resources of cloud are shared across several organizations that have some common concerns. Finally, hybrid cloud is the combination of other cloud models [4].

Due to complex and dynamic nature of cloud environment, task scheduling strategy plays an important role to utilize cloud benefits. To make good use of resources in different scales, cloud computing requires efficient task scheduling algorithm to manage them [6]. Here, "scheduling" is defined as a mechanism that determines which tasks are allocated to run on the machines of a distributed environment (see Fig. 2) [7-8].

Howbeit, cloud environment nowadays presents a better schema to perform the submitted tasks in terms of responsiveness, scalability, and flexibility, task scheduling problem is known to be NP-complete problem in cloud system [9-10].

Generally, scheduling methods take into account one or two objectives that include makespan, task priority, profit etc. The nature of cloud environment often results in energy-related issues that be investigated for increasing profit of service providers and environment protection.

The security problems usually make cloud computing formidable, particularly when infrastructure are owned by an outside party that provides services to the public. In cloud environments, delivering security service is extremely difficult in comparison with tradition computing systems since many distributed resources can result in a large attack surface.

*Corresponding author's email: najme.mansouri@gmail.com

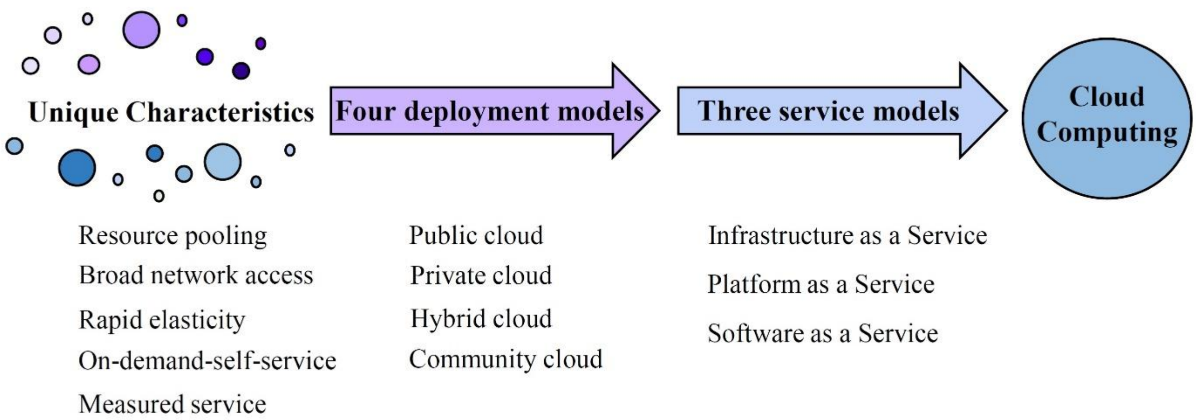

Fig. 1. Overview of cloud computing [5].

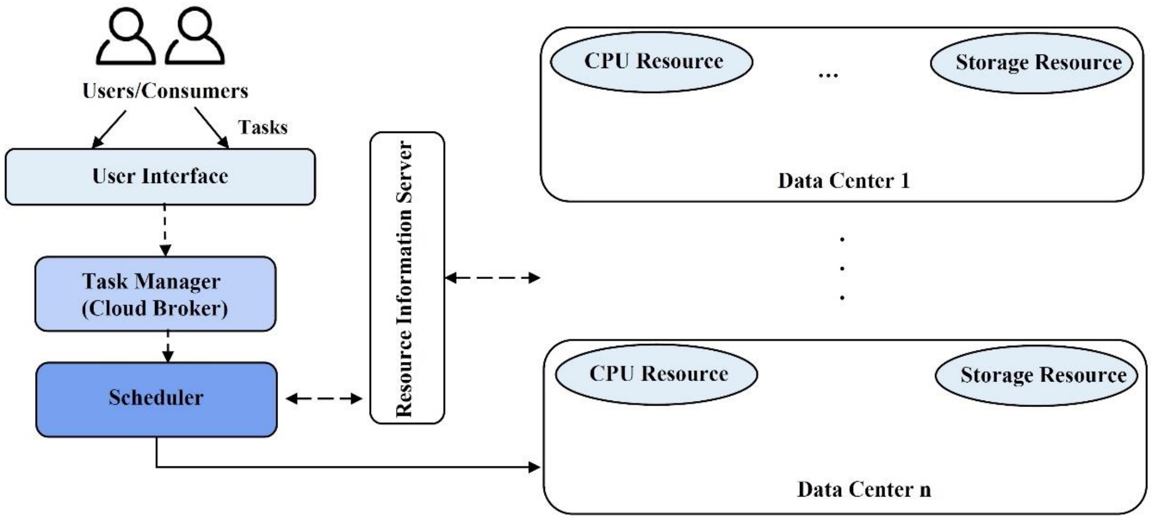

Fig. 2. Task scheduling process in cloud [11].

Consequently, security is recognized as the biggest challenge in cloud computing. Until recently, high performance has been the first goal in distributed systems, and this concern has been considered without giving much attention to security parameters [12-13].

Unfortunately, most of the task scheduling methods are unsuitable for large-scale systems because the results of these approaches are usually far from optimal. This implies that there is plenty of room to enhance tasks scheduling methods for cloud environment. In this work, we focus on resource utilization and the security of execution. We propose an Energy and Security Awareness Task (ESAT) scheduling algorithm that considers not only the security cost but also the energy consumption in a cloud system. Moreover, the fuzzy system is designed to determine an approximate rather than a precise pattern and so improves the numerical computation by using linguistic labels stipulated based on the membership functions. Finally, we evaluate the proposed ESAT algorithm through extensive simulations. Results prove that our ESAT strategy is highly effective and efficient in improving performance and can present security and efficient service for task scheduling in cloud environment.

The remainder of this paper is organized as follows. Section 2 gives a survey of related works. In Section 3, we describe the proposed task scheduling approach. Section 4 explains the evaluation of the proposed strategy and proves its benefits under different working conditions. Finally, the concluding remarks are drawn in Section 5.

2- RELATED WORKS

Security and energy consumption are the crucial components in cloud environments. Efficient scheduling method can satisfy the requirements of user and providing of services. Nevertheless, a lot of research has been performed on resource allocation, tasks scheduling remains the challenging issue for both industry and academia [14-15]. This section reviews some of the existing scheduling methods for cloud computing.

Cerero et al. [16] presented a security supportive energy-aware scheduling to enhance security in cloud environment.

The proposed strategy considers different security constraints. Each security constraint is represented based on security demands (SD) of tasks and trust levels (TL) provided by data centers. Then user can choose one of the possible security levels based on the proposed security model. Therefore, users enable to determine longer or shorter keys for cryptographic procedures based on the provided information. The results of experiments proved that the proposed model might be applied in any High-Performance Computing environment that needs the assignation of tasks to computing nodes.

Shishido et al. [17] investigated the effects of various meta-heuristics (i.e., Particle Swarm Optimization (PSO), the Genetic Algorithm (GA), and the Multi-Population Genetic Algorithm (MPGA)) techniques on a task scheduling algorithm in cloud environment. The authors applies a security-aware and cost-aware workflow scheduling algorithm [18] as a base method. The simulation results based on real scientific applications indicated that both GA and MPGA were consistently more efficient than PSO in scheduling problem.

Ismail et al. [19] presented an Energy-Aware Task Scheduling algorithm on Virtual Machine (EATSVM) to reduce energy consumption in cloud system. The introduced strategy takes into account both active and idle virtual machines to assign a virtual machine for executing a task. It also monitor the increase in the energy consumption of the running tasks on a virtual machine. The simulation results indicated that EATSVM strategy saves more energy in comparison with ECTC method [20]. Since EATSVM strategy uses the power consumption range of the virtual machines rather than the virtual machine's idle power consumption.

Zhang et al. [21] introduced a Job Security Scheduling Strategy (JSSS). The authors proposed a cloud architecture with four layers as SOA Architecture, Management Middleware, Resource virtualization, and Physical Resources. Then, they considered security demand and trust level to satisfy the security level of job scheduling. Afterwards, the proposed strategy models the scheduling problem based on Genetic Algorithm (GA). In the proposed model, each chromosome shows a schedule of a set of jobs on several data centers and then each gene is defined as a pair to indicate the relationships between jobs and data centers. The results demonstrated that the proposed scheduling algorithm cloud reduce processing time for various number of jobs.

Liu et al. [22] proposed a self-adaptive layered sleep vision-based strategy to solve secure scheduling problem in cloud environment. The authors applied a decision-making tree structure for feature classification of resource security dynamic scheduling tasks. In addition, they considered a top down analytical approach for multithread space reconstruction of resources in the storage of cloud. Finally, they proposed a self-adaptive filtering strategy for removing redundant resources. The average scheduling accuracy of the proposed strategy is about 95.5%, which is 31.7% superior to the traditional strategy (72.5%) in term of accuracy for resource security dynamic scheduling in cloud environment.

Lou et al. [23] presented a Genetic-based Task Scheduling Algorithm (GTSA), which not only reduces execution time but also guarantee load balance. The authors considered five conditions as new task arriving, long time waiting, testing resource idle, and testing resource crash and load high during task scheduling process. The results of experimental indicated that the proposed task scheduling strategy could reduce the makespan and realize the load balance.

Mansouri and Javidi [24] proposed a Cost-based Job Scheduling (CJS) algorithm for cloud environment. CJS considers the characteristics of data intensive and computation intensive tasks, simultaneously. Moreover, it models the characteristics of network such as jitter and packet loss. Finally, it defines a linear cost function based on transfer cost, network cost, and computation cost. The simulation results with CloudSim indicated that CJS cloud improve processor utilization and makespan. Nevertheless, the main weakness of CJS is the values of weights in the cost function the effect on the performance. In this paper, our proposed algorithm (ESAT) solves this problem with designing appropriate fuzzy system.

Achar et al. [25] introduced Optimal Task Scheduling Strategy (OTSS), which applies tree based data structure for execution of tasks in efficient manner. At first, the authors prioritized the tasks according to their size and Virtual Machines based on their MIPS value. Therefore, task having highest size has highest rank. They constructed a tree named Virtual Machine Tree (VMT) in which each Virtual Machine is represented with a node. Then, the proposed strategy executes the grouping of task based on the number of leaves in the VMT. The simulation results with CloudSim proved that OTSS gives better performance in comparison with other traditional scheduling algorithms such as FCFS.

To the best of our knowledge, some task scheduling methods consider single objective and attempt to reduce execution time. In dynamic and complex cloud environment, several QoS objectives should be considered for task scheduling problem. Security, makespan, and energy consumption are the crucial parameters in cloud scheduling. Therefore, we try to satisfy the user requirements and service provider by a hybrid task scheduling which uses fuzzy theory during assigning tasks to the data centers.

3- PROPOSED TASK SCHEDULING ALGORITHM

The proposed scheduling algorithm considers five important factors that noticeably affect the computation time, monetary cost, energy consumption, security level, and centrality. Fig. 3 indicates the overall system model.

3-1- Task execution

The total cost for the execution of tasks can be obtained as an average completion time of tasks on machines, to which they are assigned.

$$\mathrm{Cos}t_{Ex} = \frac{\sum_{j \in Tasks} Completion_{ij}}{\underset{\substack{j \in Tasks \\ i \in Machines}}{Max} Completion_{ij} \times n} \tag{1}$$

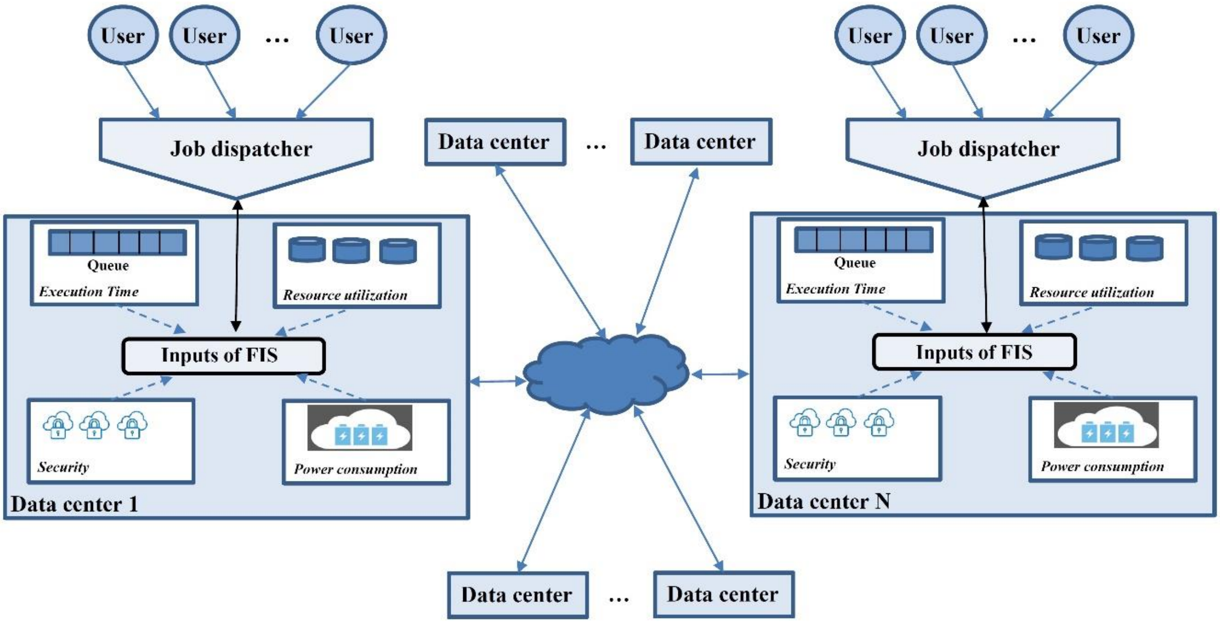

Fig. 3. The proposed system model.

Where n is number of tasks. *Tasks* and *Machines* indicate the set of total tasks and the set of total machines in the system, respectively. $Completion_{ij}$ shows the completion time of task j on machine i and is computed by Eq. (2) [26].

$$Completion_{ij} = Execution_{ji} + Ready_i \tag{2}$$

Where $Ready_i$ shows the finishing execution time of tasks that are previously assigned to the machine i. $Execution_{ji}$ is determined based on Eq. (3).

$$Execution_{ji} = \frac{L_j}{C_i} \tag{3}$$

Where L_j and C_i show the computational load of task j in millions of instructions and the computing capability of machine i in millions of instructions per second, respectively.

3-2- Resource utilization

Here, we consider the resource utilization as a portion of the average idle time of machines on which tasks are assigned. Therefore, the resource utilization is given by Eq. (4) [26].

$$Cost_{Re} = \sum_{i \in Machines} \left(1 - \frac{Completion_i}{Makespan}\right) \times Idle\_param_i \tag{4}$$

Where $Completion_i$ is the completion time of machine i and is found by Eq. (5).

$$Completion_i = Ready_i + \sum_{\substack{j \in Tasks \\ Assigned\_machine_j = i}} Execution_{ji} \tag{5}$$

In addition, Makspan is a maximum value of $Completion_i$. $Idel\_param_i$ is expressed as Eq. (6).

$$Idle\_param_i = \frac{\sum_{j \in Tasks_i} Execution_{ji}}{Completion_i} \tag{6}$$

Where, $Task_i$ indicates the set of the tasks that are executed on machine i.

3-3- Security cost

The machine failures cost is defined as the products the execution time of tasks on the machines and the failure probabilities. Hence, security cost is achieved by Eq. (7) [26].

$$Cost_{Sec} = \sum \frac{FP_{ji} \times Execution_{ji}}{(\underset{\substack{j \in Tasks \\ i \in Machines}}{Max} Execution_{ji}) \times n} \tag{7}$$

Where, FP shows the failure probabilities.

3-4- Power consumption

We apply the same energy model as in [27] for our problem as follows.

$$Cost_{Pow} = k \times P_{max} + (1-k) \times P_{max} \times U \tag{8}$$

Where, U shows the CPU utilization and k indicates the fraction of power used in the idle status. In addition, P_{max} shows the power consumption in the peak load.

3-5- Centrality

Recently, centrality metrics have gained importance in real networks with heterogeneous nature [28]. If task is data intensive then the data centers with high centrality are

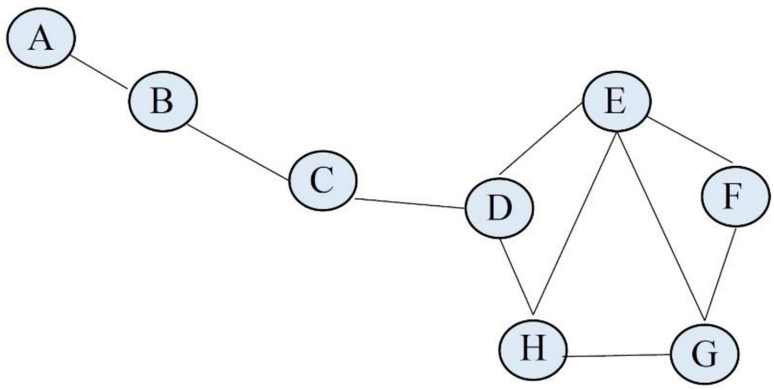

Fig. 4. An example of graph for closeness centrality calculation.

Table 1. Closeness centrality calculation for node D and node A.

Closeness centrality calculation for Node D		Closeness centrality calculation for Node A	
Node	Shortest path from D	Node	Shortest Path from A
A	3 (D-C-B-A)	B	1
B	2	C	2
C	1	D	3
E	1	E	4
F	2	F	5
G	2	G	5
H	1	H	4
$Centrality = \frac{7}{(3+2+1+1+2+2)} = 1.71$		$Centrality = \frac{7}{(1+2+3+4+5+5+4)} = 3.43$	

appropriate and ensure an improved access time. In other words, the centrality of a data center presents the measure of the relative importance of a data center in a network. There are several centrality metrics such as closeness, degree, betweenness, eccentricity, and eigenvector [28]. In this paper, we only considers the closeness centrality that is obtained based on Eq. (9). In closeness centrality measure, a node is considered important if it is relatively close to all other nodes [28].

$$Centrality(u) = \frac{n-1}{\sum_{u \neq v} d(u,v)} \tag{9}$$

Where n is the number of nodes in graph and $d(u,v)$ indicates the distance between u and v.

Consider we have 8 nodes like Fig. 4. Table 1 shows the closeness centrality calculation for node D and node A. In the closeness centrality, lower values show more central nodes. Therefore, node D is more central since node D's closeness centrality is 1.71 and node A's is 3.43.

3-6- Fuzzy system

To calculate the merit of each data center for executing task T, a fuzzy inference system will be designed. Fuzzy logic uses linguistic terms so it is close to human thinking style. It assigns membership degrees to the concepts, which express ambiguity. Then it uses the if/then rules for evaluating the various cases of each inputs fuzzy sets. Therefore, the optimum results are determined much close to the target outputs [29]. In general, fuzzy system can be found faster and smoother response than the conventional systems with less complexity. We use common fuzzy control method, namely Mamdani because of several advantages such as low overhead and intuitive for expert opinion. Then, it considers a collection of several parallel rules to describe how the fuzzy system should make a decision.

The proposed fuzzy system considers five input parameters (i.e., $Cost_{Ex}$, $Cost_{Re}$, $Cost_{Sec}$, $Cost_{Pow}$, and *Centality*) and presents *Merit* as an output output. The overall structure of the fuzzy system for determining the *Merit* of data center is presented in Fig. 5. Fig. 6 indicates the membership functions for parameters of the proposed fuzzy system that are obtained using the fuzzy logic toolbox of Matlab. We define three membership functions (i.e., *Low*, *Medium* and *High*) for all parameters. The designed fuzzy system contains 100 rules and some of them are listed in Table 2.

Finally, the proposed strategy calculates *Merit* for each task across all data centers and selects the data center that has the highest *Merit*. Fig. 7 shows the overall flow chart of ESAT algorithm.

4- EXPERIMENT RESULTS

In this section, the architecture of simulation tool and configuration are presented and finally the simulation results are given.

1-4- Simulation architecture

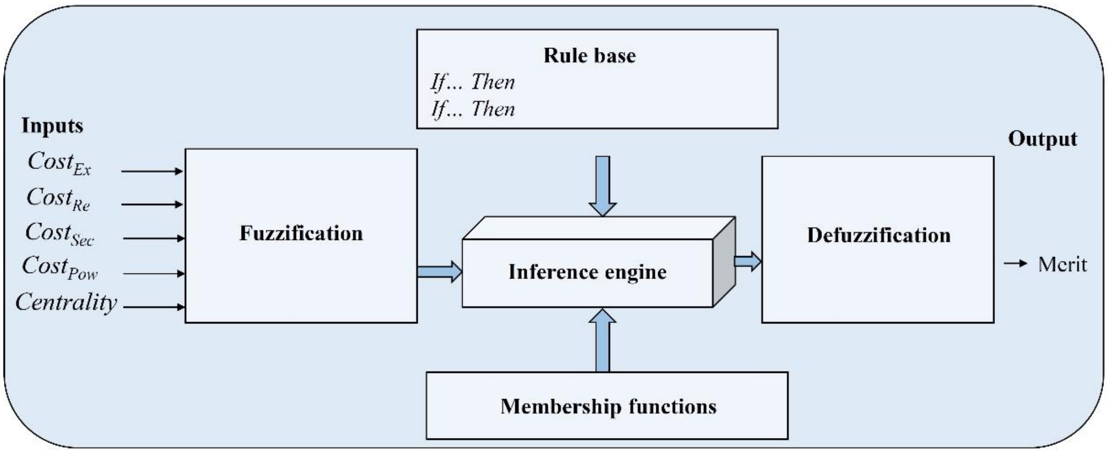

Fig. 5. The proposed fuzzy inference system (FIS).

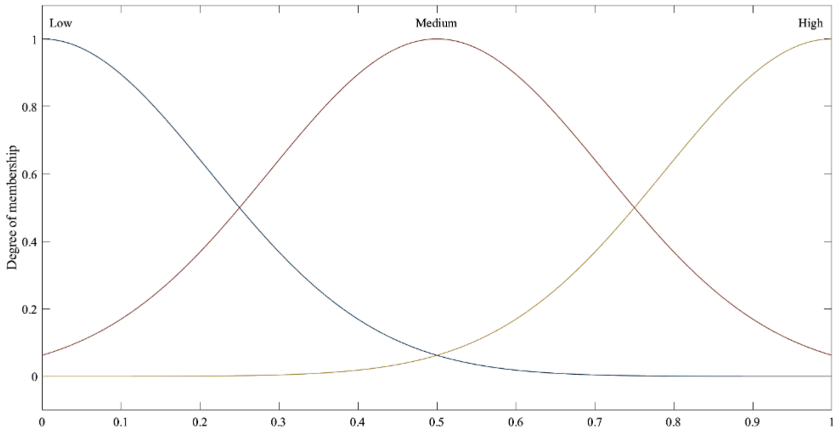

Fig. 6. Membership functions.

Table 2. Fuzzy rules of controller system.

$Cost_{Ex}$	$Cost_{Re}$	$Cost_{Sec}$	$Cost_{Pow}$	Centrality	Merit
Low	Low	Low	Low	High	High
High	High	High	Medium	Medium	Low
Low	Low	Medium	Medium	High	High
Low	Medium	High	Medium	Medium	Medium
High	High	Medium	Low	Low	Medium
High	High	High	High	Low	Low
Medium	Low	Medium	Medium	Low	Medium
Low	Low	Low	Medium	Medium	High
High	Medium	High	High	Medium	Low
Medium	Low	Low	Medium	High	High

We use CloudSim toolkit [30-31] that provides large-scale cloud computing environment, user defined allocation algorithms, and network connections. The architecture of CloudSim is presented in Fig. 8. There are three main layers: 1) User code that provides general configuration like user requirements, 2) CloudSim represents the main components such as VM provisioning and network topology, and 3) CloudSim core simulation engine that supports Queuing and the processing of events.

2-4- Configuration

In our simulation, each task is arrived according to the Poisson distribution after the previous task and each tasks needs 1 ~ 5 files. Table 3 indicates the simulation parameters.

3-4- Performance evaluation

Table 3. Simulation parameters.

Parameters	Value
Number of data centers	10-40
Total number of VMs	70
MIPS of processing element	500-2500
Number of processing element per VM	1-5
VM memory (RAM)	512-2048 (MB)
Total number of tasks	100-500
Length of task	100-500

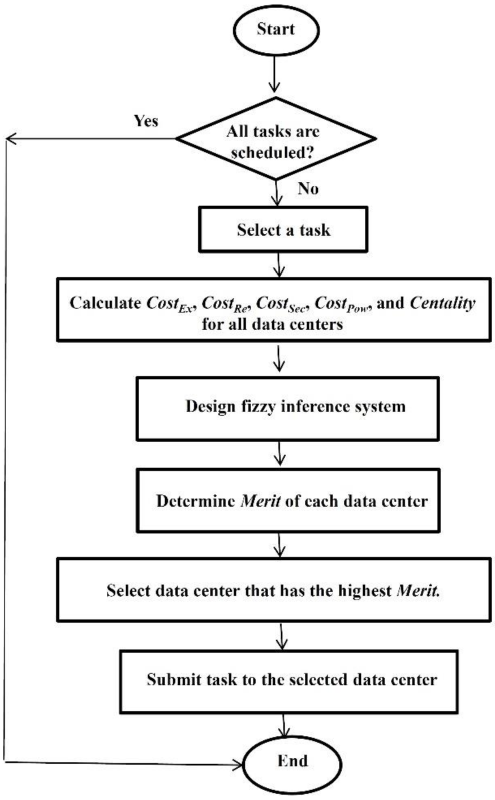

Fig. 7. Flow chart of ESAT algorithm.

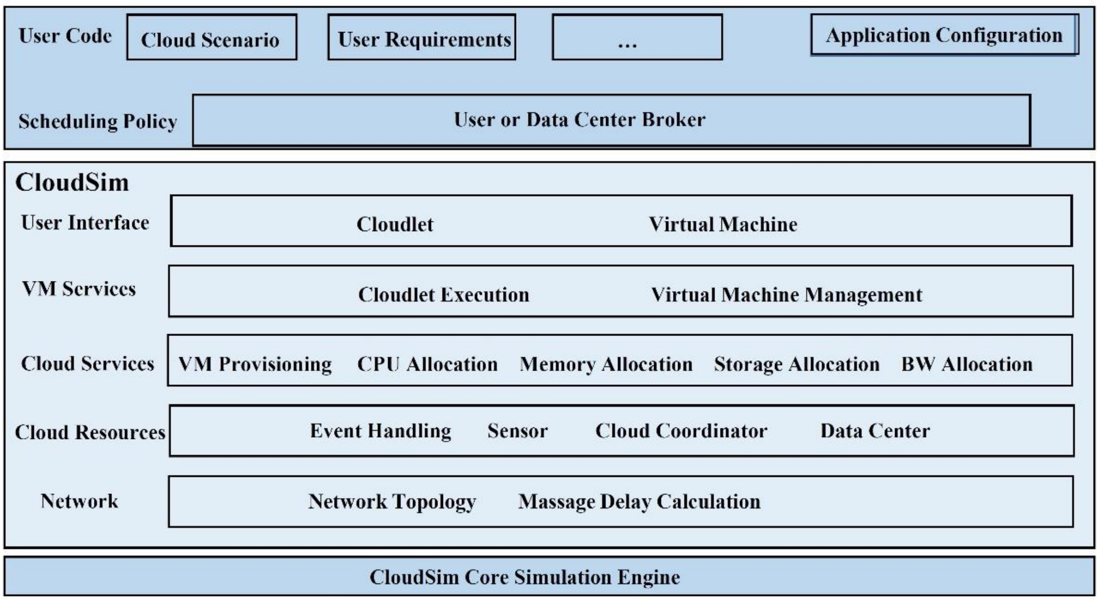

Fig. 8. CloudSim Architecture [30].

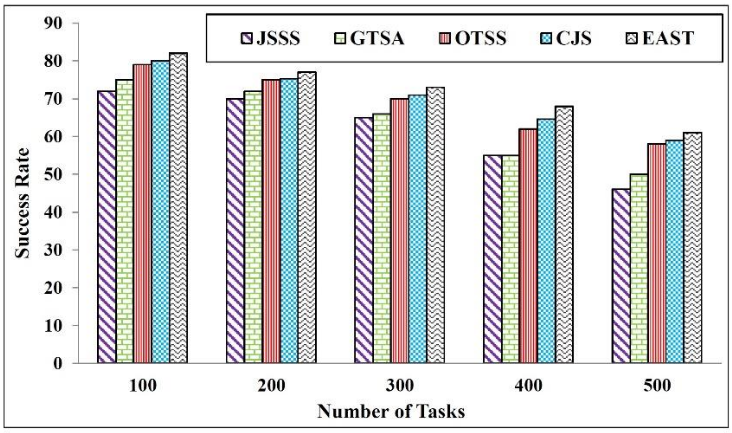

Fig. 9. Success rate for different number of tasks.

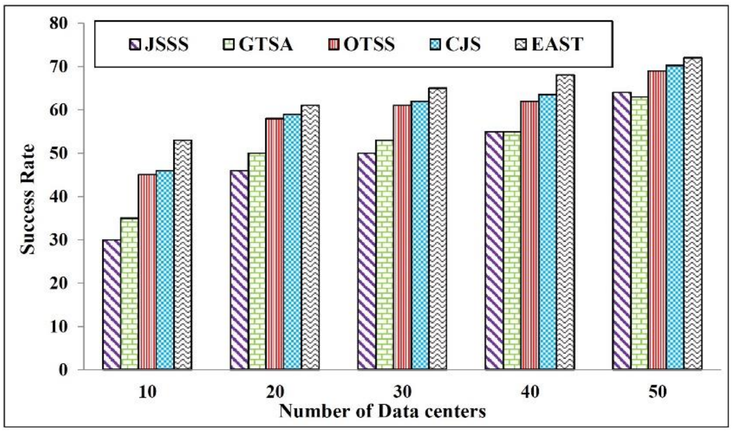

Fig. 10. Success rate for different number of data centers.

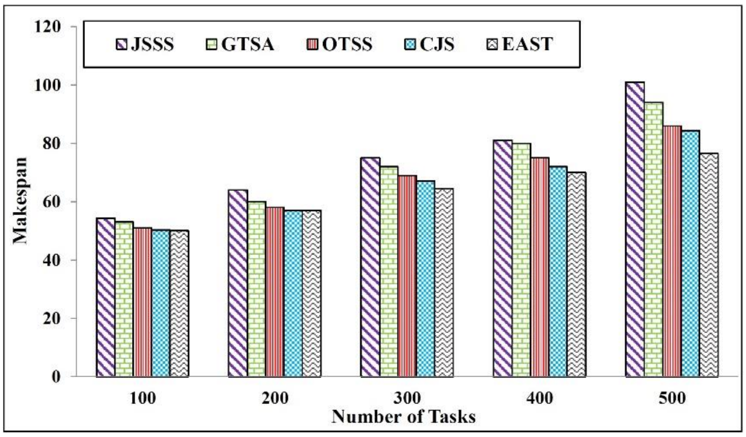

Fig. 11. Makespan for different number of tasks.

Firstly, we study the success rate that is defined as the ratio of the number of tasks executed successfully to the total number of tasks submitted to the system. Fig. 9 shows the success rate for different number of tasks. We can see that EAST algorithm gives high success rate to other scheduling algorithms since it considers security and cost factors, simultaneously. Fig. 10 indicates the success rate for different number of data centers. It is obvious that when number of data center is greater, the resources are sufficient in system so all algorithms represent near performance. For 10 data centers, EAST algorithm increases success rate about 40% in comparison with JSSS strategy.

The makespan has been evaluated as the most common performance metric by a majority of scheduling approaches. The highest finishing time among all tasks is defined as makespan. Overall, as we can observe from Fig. 11, EAST algorithm has better makespan as compared to OTSS, GTSA, and JSSS. As the number of tasks increases in data centers,

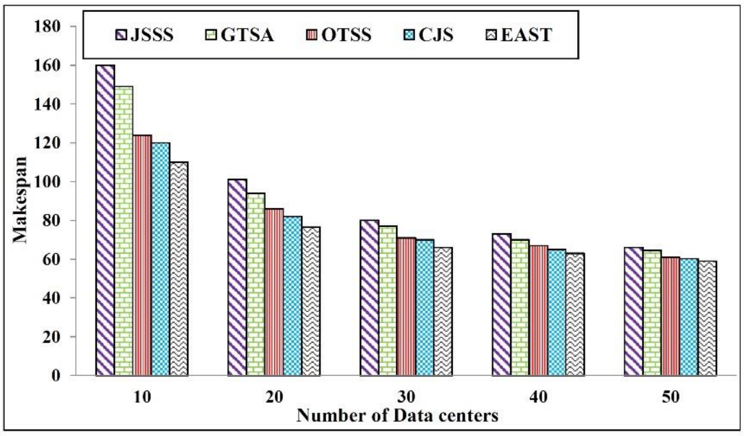

Fig. 12. Success rate for different number of data centers.

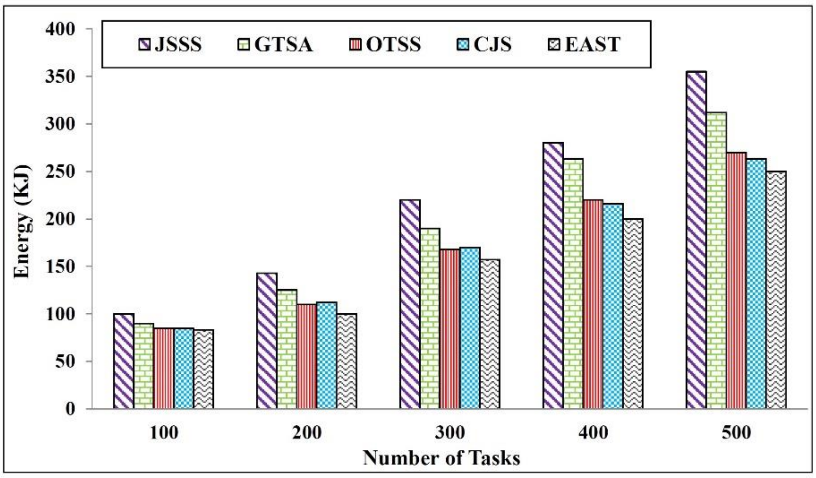

Fig. 13. Energy consumption for different number of tasks.

EAST outperforms other algorithms greatly. Fig. 12 illustrates makespan for different task scheduling method with various number of data centers. We can observe from Fig. 12 that EAST strategy has lower makespan compared to the OTSS strategy (about 11%).

The main reason is that EAST schedules tasks based on the characteristic of data centers and the length of tasks. Therefore, it does not waste powerful resources on tasks with low computation requirement. It is noteworthy to mention that in Fig. 12, CJS reduces makespan about 14% compared with JSSS. The reason for this is that CJS takes into account the data intensive and computation intensive aspects of the task during scheduling procedure.

Fig. 13 illustrates the total energy consumption for different number of tasks. We can see two results: 1) the energy consumption increases with the increasing of number of tasks and 2) the proposed scheduling algorithm shows the lower energy consumption compared to other methods. For example, EAST reduces energy consumption about 6% compared with CJS. This is because EAST considers energy cost during assigning tasks to the data centers.

Fig. 14 reports the waiting time that is computed as the time that a task waits from its submission to completion in the queues. From the results, it is obtained that the EAST algorithm reduces the waiting time by an average of 27% when compared with JSSS and by an average of 14% when compared with GTSA. This can be related to EAST ability in selecting the data canter with low ready time. Fig. 15 shows the waiting time of task scheduling algorithms for different number of data centers. With decreasing the number of data centers, the increasing trend of the waiting time by the proposed scheduling method is significantly less than the other methods. Since EAST algorithm considers centrality parameter during scheduling process and hence data retrieval time is improved.

Fig. 16 presents the graphical representation of number of tasks verses the obtained degree of imbalanced. Degree of imbalance is defined by Eq. (10).

$$D_i = \frac{Leng\_Tasks}{Num\_CPU \times CPU\_MIPS} \tag{10}$$

Where *Leng_Tasks* indicates the total length of tasks that are assigned to VM_i, *Num_CPU* shows the number of CPU

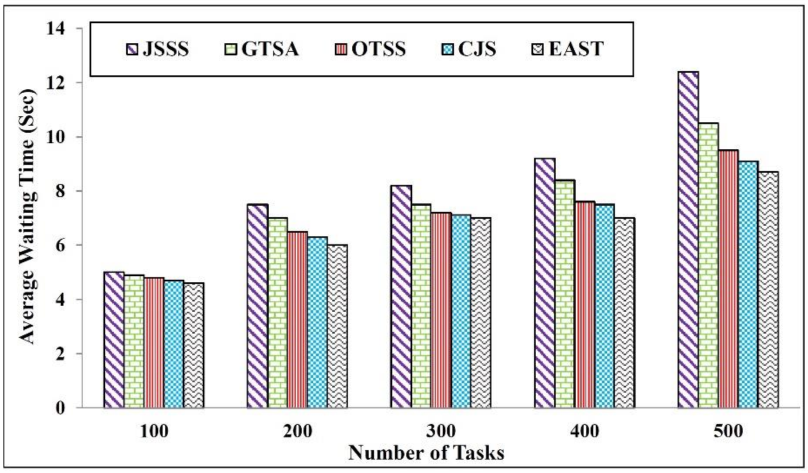

Fig. 14. Average wiating time for different number of tasks.

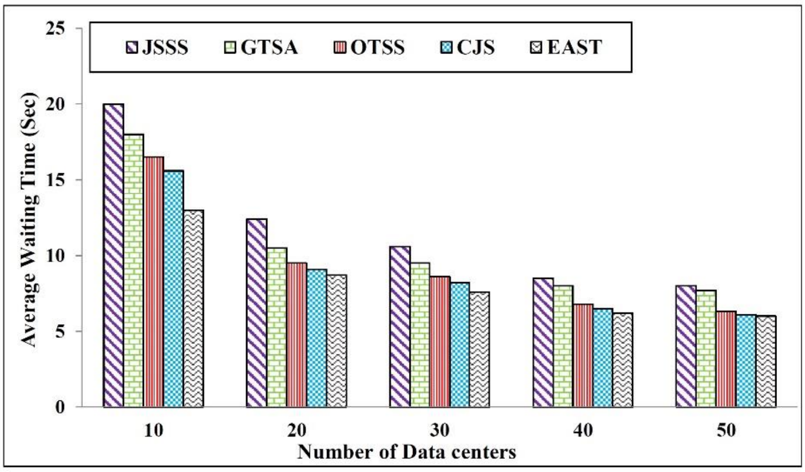

Fig. 15. Average wiating time for different number of data centers.

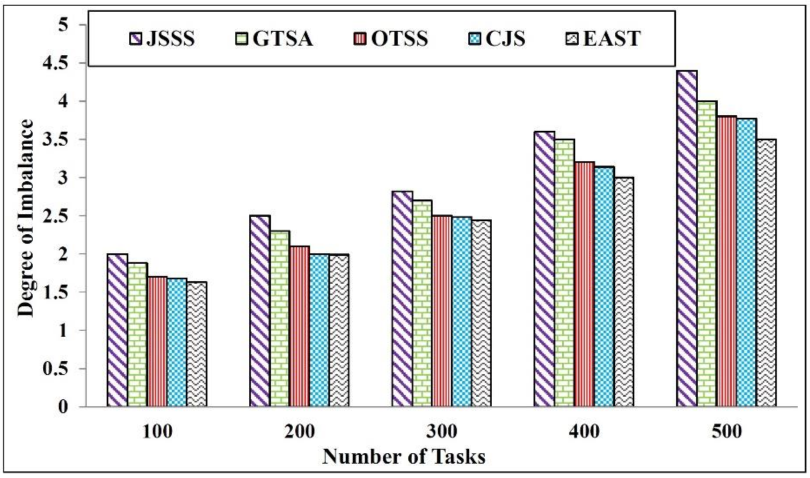

Fig. 16. Degree of imbalance for different number of tasks.

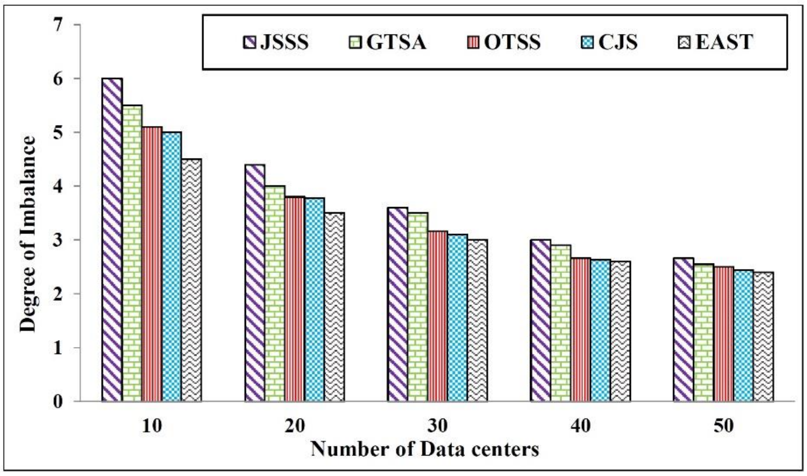

Fig. 17. Degree of imbalance for different number of data centers.

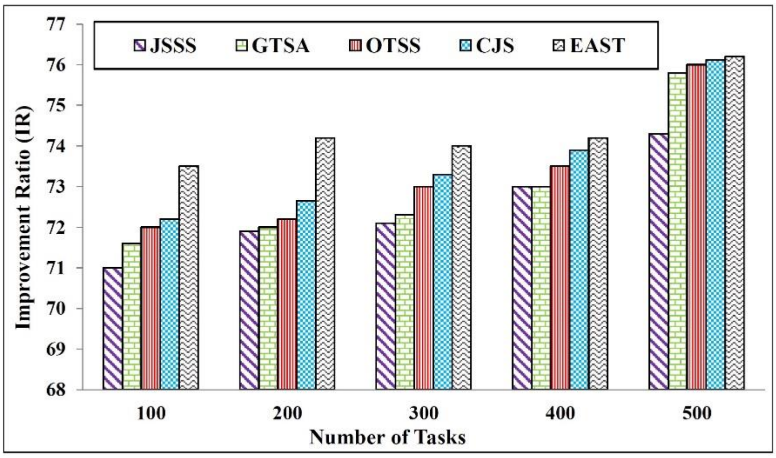

Fig. 18. Improvement ratio for different number of tasks.

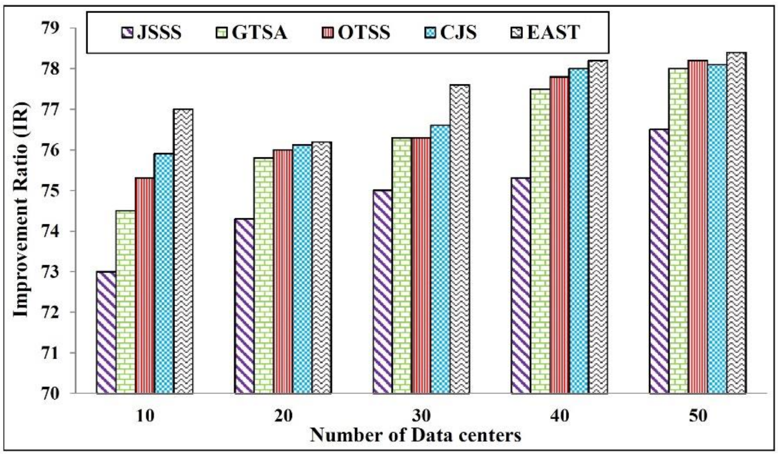

Fig. 19. Improvement ratio for different number of data centers.

and *CPU_MIPS* is the capability of CPU.

We can see that EAST strategy improved the degree of imbalance by an average of 20% when compared with JSSS and by an average of 13% when compared with GTSA. Because EAST strategy considers the variation of computing capability of machine and the pool of processing cycles.

According to Fig. 17, it is clear that our proposed EAST strategy consistently outperforms other scheduling algorithms in the degree of imbalance parameter. Since tasks are assigned to data centers according to resource utilization.

Fig. 18 explains the improvement ratio that presents the efficiency of algorithm according to the execution time reduction. Hence, it is obtained by Eq. (11) [31]:

$$IR_j\% = \frac{\sum_{i=1, i\neq j}^{n} ExT_i - ExT_j}{\sum_{i=1, i\neq j}^{n} ExT_i} \times 100 \tag{11}$$

Where ExT_i shows the execution time of ith algorithm. Form Fig. 18 and Fig. 19, we can see that EAST algorithm has the best improvement ratio in comparison with JSSS, GTSA, and OTSS algorithms. These results are because the proposed algorithm results in lower execution time by considering important parameters such as centrality and data center features. Moreover, it is obvious that EAST algorithm gains better performance than other methods under the heavy loads.

5- CONCLUSION

Cloud redefines the security problems targeted on task scheduling. It is critical to study the cloud task management with security implications and extend scheduling model to multidimensional decision. In this paper, we develop EAST approach that takes into account the heterogeneity of the cloud as well as the security and energy consumption while assigning a new task. In addition, we design fuzzy inference system to enhance EAST performance and provide load balancing. A comparative analysis with other strategies (i.e., JSSS, GTSA, OTSS, and CJS) demonstrated that it had good balanced utilizations and good time saving compared to other scheduling algorithms. Future work should also pursue the optimization of security operations. In addition, we want to use game theory for the optimization of the trust levels of data centers.

REFERENCES

[1] B. Shojaiemehr, A.M. Rahmani, N. Nasih Qader, Cloud computing service negotiation: A systematic review, Computer Standards & Interfaces, vol. 55, pp. 196-206, 2018.

[2] N. Mansouri, M. M. Javidi, B. Mohammad Hasani Zade, Using data mining techniques to improve replica management in cloud environment, Soft Computing, pp. 1-26, 2019.

[3] N. Mansouri, M.M. Javidi, A new prefetching-aware data replication to decrease access latency in cloud environment, Journal of Systems and Software, vol. 144, pp. 197-215, 2018.

[4] B.R. Nadh Singh, B.Raja Srinivasa Reddy, A review on big data mining in cloud computing, Innovations in Computer Science and Engineering, pp. 131-142, 2017.

[5] A. Ragmani, A. El Omri, N.Abghour, K. Moussaid, M. Rida, An intelligent scheduling algorithm for energy efficiency in cloud environment based on artificial bee colony, In: 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), 2017.

[6] A.R. Arunarani, D. Manjula, V. Sugumaran, Task scheduling techniques in cloud computing: A literature survey, Future Generation Computer Systems, vol. 91, pp. 407-415, 2019.

[7] L. Yuanjun, L. Sisi, T. Diyin, Multi-phase integrated scheduling of hybrid tasks in cloud manufacturing environment, Robotics and Computer-Integrated Manufacturing, vol. 61, 2020.

[8] N. Mansouri, G.H. Dastghaibyfard, A. Horri, A novel job scheduling algorithm for improving data grid's performance, In: International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, (2011).

[9] S. Srichandana, T. Ashok Kumar, S. Bibhudatta, Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm, Future Computing and Informatics Journal, vol. 3, pp. 210-230, 2018.

[10] N. Mansouri, Network and data location aware approach for simultaneous job scheduling and data replication in large-scale data grid environments, Frontiers of Computer Science, vol. 8 (3), pp. 391-408, 2014.

[11] M. Roshni Thanka, P. Uma Maheswari, E. Bijolin Edwin, An improved efficient: Artificial Bee Colony algorithm for security and QoS aware scheduling in cloud computing environment, Cluster Computing, vol. 22, pp.10905–10913, 2019.

[12] F. Abazari, M. Analoui, H. Takabi, S. Fu, MOWS: Multi-objective workflow scheduling in cloud computing based on heuristic algorithm, Simulation Modelling Practice and Theory, vol. 93, pp. 119-132, 2018.

[13] N. Mansouri, M.M. Javidi, A hybrid data replication strategy with fuzzy-based deletion for heterogeneous cloud data centers, The Journal of Supercomputing, vol.74 (10), pp. 5349-5372, 2018.

[14] A. Wilczyński, J. Kołodziej, Modelling and simulation of security-aware task scheduling in cloud computing based on Blockchain technology, Simulation Modelling Practice and Theory, vol. 99, 2020.

[15] N. Mansouri, M. M. Javidi, Cost-based job scheduling strategy in cloud computing environments, Distributed and Parallel Databases, 2019.

[16] D. Fernández-Cerero, A. Jakóbik, D. Grzonka, J. Kołodziej, A. Fernández-Montes, Security supportive energy-aware scheduling and energy policies for cloud environments, Journal of Parallel and Distributed Computing, vol. 119, pp. 191-202, 2018.

[17] H. Yoshikazu Shishido, J. Cezar Estrella, C.F. Motta Toledo, M. Silva Arantes, Genetic-based algorithms applied to a workflow scheduling algorithm with security and deadline constraints in clouds, Computers and Electrical Engineering, vol. 69, pp. 378-394, 2018.

[18] Z. Li, J. Ge, H. Yang, L. Huang, H. Hu, H. Hu, B. Luo, A security and cost aware scheduling algorithm for heterogeneous tasks of scientific workflow in clouds, Future Generation Computer Systems, vol. 65, pp. 140-52, 2016.

[19] L. Ismail, H. Materwala, EATSVM: Energy-aware task scheduling on cloud virtual machines, Procedia Computer Science, vol. 135, pp. 248-258, 2018.

[20] Y.C. Lee and A. Y. Zomaya, Energy efficient utilization of resources in cloud computing systems, Journal of Supercomputing, vol. 60, pp. 268-280, 2012.

[21] H. Zhang, Research on job security scheduling strategy in cloud computing model, International Conference on Intelligent Transportation, Big Data & Smart City, pp. 649-652, 2015.

[22] X. Liu, Y. Zhou, A self-adaptive layered sleep-based method for security dynamic scheduling in cloud storage, 4th International Conference on Information Science and Control Engineering, pp. 99-103, 2017.

[23] Y. Lou, T. Zhang, J. Yan, K. Li, Y. Jiang, H. Wang, J. Cheng, Dynamic scheduling strategy for testing task in cloud computing, Sixth International Conference on Computational Intelligence and Communication Networks, pp. 633-636, 2014.

[24] N. Mansouri, M. M. Javidi, Cost-based job scheduling strategy

in cloud computing environments, Distributed and Parallel Databases, pp. 1-36, 2019.
[25] R. Achar, P. Santhi Thilagam, D. Shwetha, H. Pooja, R. Andrea, Optimal scheduling of computational task in cloud using virtual machine tree, Third International Conference on Emerging Applications of Information Technology (EAIT), pp. 143-146, 2012.
[26] J. Kołodziej, F. Xhafa, Meeting security and user behavior requirements in grid scheduling, Simulation Modelling Practice and Theory, vol. 19, pp. 213-226, 2011.
[27] A. Beloglazov, J. Abawajy, R. Buyya, Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing, Future Generation Computer Systems, vol. 28, pp. 755-768, 2012.
[28] M. Ali, K. Bilal, S.U. Khan, B. Veeravalli, K. Li, A.Y. Zomaya, DROPS: division and replication of data in cloud for optimal performance and security, IEEE Transactions on Cloud Computing, vol. 6, pp. 305-315, 2018.
[29] N. Mansouri, M.M. Javidi, A hybrid data replication strategy with fuzzy-based deletion for heterogeneous cloud data centers, Journal of Supercomputing, vol. 74 (10), pp. 5349–5372, 2018.
[30] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A.F. De Rose, R. Buyya, CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Software – Practice and Experience, vol. 41, pp. 23-50, 2011.
[31] N. Mansouri, Adaptive data replication strategy in cloud computing for performance improvement, Frontiers of Computer Science, vol. 10 (5), pp. 925-935, 2016.
[31] N. Mansouri, B. Mohammad Hasani Zade, M.M. Javidi, Hybrid task scheduling strategy for cloud computing by modified particle swarm optimization and fuzzy theory, Computers & Industrial Engineering, vol. 130, pp. 597-633, 2019.

HOW TO CITE THIS ARTICLE

N. Mansouri, Behnam M. Hasani Zade, M.M. Javidi, Energy and Security Awareness Task Scheduling based on Fuzzy System in Cloud Computing, AUT J. Model. Simul., 52(1) (2020) 129-142.

DOI: 10.22060/miscj.2020.17354.5180