Computer Systems Science & Engineering
DOI:10.32604/csse.2022.021816
Article

Tech Science Press

Cost Effective Optimal Task Scheduling Model in Hybrid Cloud Environment

M. Manikandan[1,*], R. Subramanian[2], M. S. Kavitha[3] and S. Karthik[3]

[1]Department of Computer Science & Engineering, Sri Krishna College of Engineering and Technology, Coimbatore, 641008, India
[2]Department of Electrical and Electronics Engineering, SNS College of Technology, Coimbatore, 641035, India
[3]Department of Computer Science & Engineering, SNS College of Technology, Coimbatore, 641035, India
*Corresponding Author: M. Manikandan. Email: manimmk9792@gmail.com

Received: 15 July 2021; Accepted: 18 August 2021

Abstract: In today's world, Cloud Computing (CC) enables the users to access computing resources and services over cloud without any need to own the infrastructure. Cloud Computing is a concept in which a network of devices, located in remote locations, is integrated to perform operations like data collection, processing, data profiling and data storage. In this context, resource allocation and task scheduling are important processes which must be managed based on the requirements of a user. In order to allocate the resources effectively, hybrid cloud is employed since it is a capable solution to process large-scale consumer applications in a pay-by-use manner. Hence, the model is to be designed as a profit-driven framework to reduce cost and make span. With this motivation, the current research work develops a Cost-Effective Optimal Task Scheduling Model (CEOTS). A novel algorithm called Target-based Cost Derivation (TCD) model is used in the proposed work for hybrid clouds. Moreover, the algorithm works on the basis of multi-intentional task completion process with optimal resource allocation. The model was successfully simulated to validate its effectiveness based on factors such as processing time, make span and efficient utilization of virtual machines. The results infer that the proposed model outperformed the existing works and can be relied in future for real-time applications.

Keywords: Cost effectiveness; hybrid cloud; optimal task scheduling; virtual machine; resource allocation; make span

1 Introduction

Cloud Paradigm has been defined by the International Organization of Standardization as follows; "Cloud is a paradigm for providing network access to a reliable and resilient pool of shared resources with demand based self-service provisioning and management". Cloud network comprises of various responsibilities such as cloud protocols and activities, service types, categories and deployment models [1]. Among these, cloud deployment models are as follows; a Private Cloud, a Public Cloud, a Community Cloud and a Hybrid Cloud.

Cloud computing framework enables a user to access and leverage the resources without possessing the infrastructure. As per Service Level Agreement (SLA), the consumers are provided with access to the resources. Furthermore, resource management process is executed with the help of virtualization-based architectures [2]. In cloud computing phenomenon, the satisfaction of consumers is measured based on the Quality of Service (QoS), a parameter that defines the agreement between cloud users and Cloud Service Providers (CSP) [3]. In case of an increase in the number of cloud users, it is necessary to find an effective solution for scheduling the receiving tasks, under the norms of SLA. Moreover, task scheduling is a cloud computing technique to map the required resources or Virtual Machines (VMs) so as to execute the tasks. A sample scenario is presented in Fig. 1 in which many users are requesting for cloud resources with different SLAs. These requests have to be processed by allocating cloud resources in a simultaneous manner.

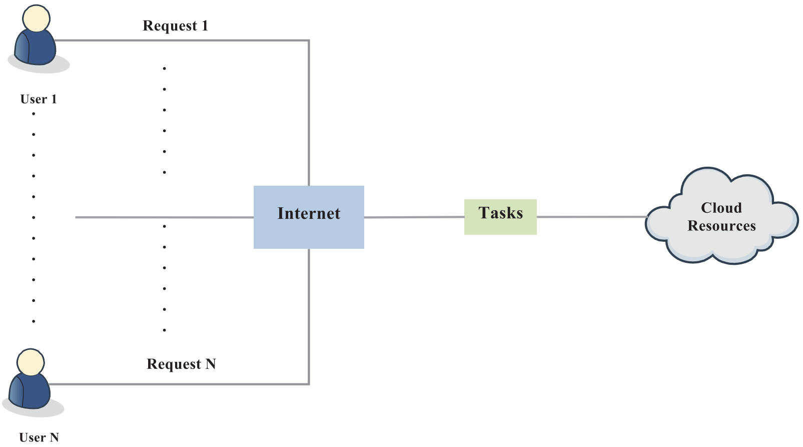

Figure 1: Execution of multiple user tasks in cloud

For example, some users may raise a request for additional storage while another user may request to increase the CPU time for performing complex tasks effectively. So, there is a demand exists for developing efficient task scheduling model. Moreover, when private cloud is considered, the resources are limited, inefficient for big data processing and cost-consuming [4,5]. Hence, the current research work is concentrated on hybrid cloud model for effective big data processing. When developing an efficient task scheduling model, the following things are primarily considered,

- Multiplicity of tasks submitted by cloud users
- Heterogeneity of resources available in the cloud
- Differences in QoS

To effectively resolve the issues, Cost-Effective Optimal Task Scheduling Model (CEOTS) is proposed and developed in current research work. Here, queuing model is determined based on the available resources and type of tasks. When the tasks are entered in a queue in parallel resource pool, scheduling is processed to satisfy the user's demands. The major contributions of the proposed model are listed below.

- Analysis of hybrid cloud model characteristics to frame the workflow
- Target based Cost Derivation is processed to minimize the cost by considering the make span
- Cross-over and mutation techniques are utilized for optimal resource management and scheduling
- Extensive experimentation was conducted to validate the effectiveness of the proposed model
- Comparison graphs are provided in the paper to establish that the proposed model is cost effective than the existing models.

Rest of the sections in this paper are organized as given herewith. Section 2 contains the description about existing job scheduling models in cloud. Problem definition is provided in section 3. Section 4 explains the processes involved in the proposed model. The results and comparisons are given in section 5. Both conclusion and future enhancement are provided in section 6.

2 Related Works

Researchers developed an extensive number of techniques to resolve the make span issues and make cloud computing service, a cost-effective one. In literature [6], a hybrid model is presented out of two optimization models to enhance the throughput of a defined cloud model. Further, an informative and valuable survey work was presented in the study conducted earlier [7]. This research work reviewed different existing resource allocation models and challenges faced during resource allocation in cloud. Another literature survey was presented in [8] which described about resource allocation with different resources, comprising data, services and cloud-based applications. An in-depth explanation was given about cloud management-based resource allocation process. Further, static and dynamic resource allocation strategies were explained in the study conducted earlier [9]. All possible task scheduling issues and mitigation techniques were analyzed in this work. Resource management and task scheduling models in cloud computing were effectively discussed in [10]. The main intention behind the development of this model is to execute the resource allocation in cloud so as to make cloud solutions, a cost-effective one. Different performance evaluations were conducted for the model and the results of the model showed a consistent increase in the performance [11]. In the study conducted earlier [12], Resource Allocation System (RAS) was developed to optimize efficient resource management in cloud.

An efficient mechanism called Heuristic Algorithm-based Deadline Early Tree (DET) model was proposed in the literature [13] to reduce the cost as well as make span. On the contrary to [13], the researchers published an article [14], which studied about the problem of reducing make span based on the cost limitation for cloud operations. To resolve the issues, user- and cost-based workflow scheduling models called Scale star were proposed to allocate the jobs to VMs. The model [15] designed a workflow schedule framework for Map Reduce tasks in heterogeneous cloud, on the basis of budget limitations and based on the concern that multiple cloud providers can provide heterogeneous cloud in parallel manner. Further, a cost minimization-based workflow scheduling model was proposed using Discrete Particle Swarm Optimization (DPSO) model for multiple cloud environment [16].

Budget pre-assignment model was proposed to minimize the processing cost for cloud operations. Moreover, in literature [17–19], the researchers discussed about Heterogeneous Earliest Finish Time (HEFT) model for optimization of cost and make span in collaborative cloud model. Fuzzy-based sorting model, with HEFT, was utilized to optimize the cost incurred in providing Infrastructure-as-a-Service over cloud [17]. Further, Particle Swarm Optimization (PSO) was integrated with HEFT in the study conducted earlier [19] for cost and make span optimization in cloud. Though the models provided effective results in public cloud, it was not feasible for hybrid cloud. Deadline-driven resource allocation model was proposed earlier in hybrid clouds and the model was implemented in Aneka Cloud Platform. This model effectively reduced the processing time [20]. In order to minimize the execution cost, least cost per connection model was proposed in the literature [21].

The authors [22] clearly defined the cost minimization problem of data processing in hybrid cloud. Further, in literature [23], the researchers described a PSO-based resource management model for hybrid cloud frameworks. Deadline-constrained workflow-based job allocation model was developed in [24]. However, this model was not considered and processed for cost and make span effectiveness in parallel manner. The aforementioned models have effectively discussed about resource management and task scheduling models. However, the issues on cost and make span have not been effectively discussed or explored in hybrid cloud models. So, the current study aims at fulfilling this research gap.

3 Problem Definition

Quality of Service (QoS) is cloud is primarily measured by two important factors such as cost effectiveness and make span [25–27]. In this background, the problem definition for the proposed model is as follows,

i) Cost-effective optimal task scheduling model is proposed here in which both cost effectiveness and make span are considered in hybrid cloud

ii) Target-based Cost Derivation (TCD) model is framed for hybrid clouds

iii) Multi-intentional task completion process, with optimal resource allocation, is derived in current work for enhanced QoS in cloud model

4 Working Process of the Proposed Model

The main objective of the proposed model is to minimize the cost and make span to ensure efficient resource allocation in cloud. Target-based Cost Derivation (TCD) model is developed in the proposed work. To resolve the issues over cost effectiveness in hybrid cloud, generic algorithm-based resource allocation is performed. The process includes three phases as follows.

- Chromosome Encoding
- VM analysis and selection
- Cross Over
- Mutation

4.1 Chromosome Encoding

The chromosome encoding for task scheduling is framed as follows.

$$TS = \left(task_{priority},\ Allocate_{resources},\ virtualmachine_{type}\right) \tag{1}$$

where, '$task_{priority}$' is the priority of tasks in task_queue, $Allocate_{resources}$ presents the priority list of resource allocation and $virtualmachine_{type}$ is the type of virtual machine. Further, 'TS' in (1) states the task scheduler. The process is clearly defined using Fig. 2.

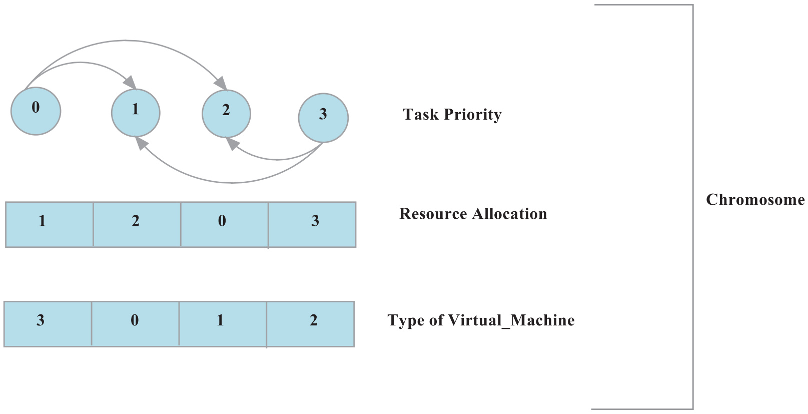

Figure 2: Chromosome encoding for resource allocation

Here, the chromosomal set is considered as a population to determine the solution set. At first, Population_set is initialized and the steps are explained herewith.

i) ‘A’ denotes the number of tasks and ‘B’ denotes the number of virtual machines in hybrid cloud whereas B_r represents the virtual machines present in private cloud.
ii) $task_{priority}$ is analyzed to satisfy the requirements of task dependency.
iii) Initiation for task to virtual machine assignment.
iv) To ensure the population differences, a random number is selected from 0 to 1.
v) When random number is less than 0.5, the values of $Allocate_{resources}$ are set as zero.
vi) And, each task is allocated to virtual machine, V_0.
vii) The non-allocated virtual machine, $virtualmachine_{type}$ is initialized and set as the private cloud.

4.2 VM Analysis and Selection

Fitness function is computed to determine the efficient VM among the population of numerous virtual machines for task allocation in hybrid cloud. In Genetic Algorithm-based models, fitness function computation is highly important since it influences the convergence speed of the model for deriving optimal solutions. To be specific, fitness function is derived based on the model with an objective for enhancing cost effectiveness. Hence, in the proposed model, fitness function is given as follows.

$$Fitness\_Rate(\alpha) = \begin{cases} processing\ cost, \ makespan \leq deadline \\ cost + p \times (makespan - deadline), makespan > deadline \end{cases} \quad (2)$$

where, ‘p’ denotes the penalty factor and ‘α’ represents the chromosome. Based on the above equation, it can be defined that when execution time of a chromosome does not exceeds the deadline, the fitness rate can be set based on the processing cost. Otherwise, the rate can be computed as a sum of processing cost and penalty factor based on the processing delay. Once the computation is completed, the chromosomes in the population set are arranged based on their *Fitness_Rate*. Further, the proposed model selects a fixed rate of population with a minimal *Fitness_Rate* while the others are rejected.

4.2.1 Expected Processing Time (EPT) Computation

Expected Processing Time, termed as EPT, can be defined as the time taken for cloud resource ‘R’ to process the task, T. EPT can also be termed as burst time for tasks. While the equation is as follows.

$$EPT_{ij} = \min_{i=1\ to\ n, j=1\ to\ m} \left(T_1 * R_j\right) \quad (3)$$

4.2.2 Task Completion Time Calculation

Task_completion time is calculated here by determining the available cloud resources. And, the expected task_completion Time CT_{ij} is derived from the summation of EPT and VM-ready time (VM_{RT}). Further, the derivation is presented in (4).

$$CT_{ij} = EPT_{ij} + VM_{RTj} \quad (4)$$

Based on the above computations, Fitness Function (FF) for efficient resource allocation of the tasks is processed here and the equation is given as (5).

$$FF = \sum_{i=1}^{n} \sum_{j=1}^{m} EPT_{ij} \quad (5)$$

where, ‘n’ is the number of tasks and ‘m’ is the available virtual resources from hybrid cloud.

4.3 Crossover Operations

In the proposed model, single-point crossover operations are carried out to determine the new chromosome and the process is explained in algorithm given below. The initial process of the algorithm randomly selects the position, 's' that lies between $[0,\ n-1]$. Followed by, the fragments 'u' and 'v' are swapped to generate the new one, and the dependency factors are not violated during the course of process. Moreover, crossover operations are processed between $Allocate_{resources}$ and $virtualmachine_{type}$. And, the algorithm is presented below.

```
    Begin
    Initialize no.of tasks $\rightarrow A$
    $\rho_1$ and $\rho_1 \leftarrow$ two chromosomes
    Initialize mid point $k \leftarrow 0$
    Select position 's'
    While $k \leq s$ do
     Perform type selection $(k,\ s,\ \rho_1,\ \rho_2)$
     Swap $\rho_1 Allocate_{resources}\ [k]$ and $\rho_2 Allocate_{resources}\ [k]$
     $k++$
    Select Type $(k,\ s,\ \rho_1,\ \rho_2)$
$\rho_1.Allocate_{resources}\ [k] = r$
$\rho_2.Allocate_{resources}\ [k] = r$
$\rho_1.virtualmachine_{type}\ [r] = type_1$
$\rho_2.virtualmachine_{type}\ [r] = type_2$
    If $\exists i \geq s \& (\rho_2.Allocate_{resources}\ [i] \neq r)$ then
$\rho_2.virtualmachine_{type}\ [r] = type_1$
    Else if $type_1 \neq type_2$ then
$\rho_2.virtualmachine_{type}\ [r] = random\ (type_1,\ type_2)$
    $num\ (v_j) \leftarrow virtual\ machines$ in private cloud
    $C = count_j$
    If $C > num_j$ then
    $l = C - num\ (v_j);$
    Random selection of virtual machines
    End
```

As shown in the algorithm, the operations are carried out based on the number of tasks in workflow queue and population, ρ_1 *and* ρ_2. When an appropriate virtual machine of a particular job t is positioned at (i), then the particular type is assigned for task completion. Additionally, the virtual machine gets mutated and the new type is ensured not to be equal alike the previous one. The algorithm further solves the conflict that occurs on the requirement of virtual machines, accessible in the private cloud. An efficient virtual machine is selected from the list for processing the tasks.

4.4 Mutation Operations

This section describes the process of mutation and the process is shown in Fig. 3.

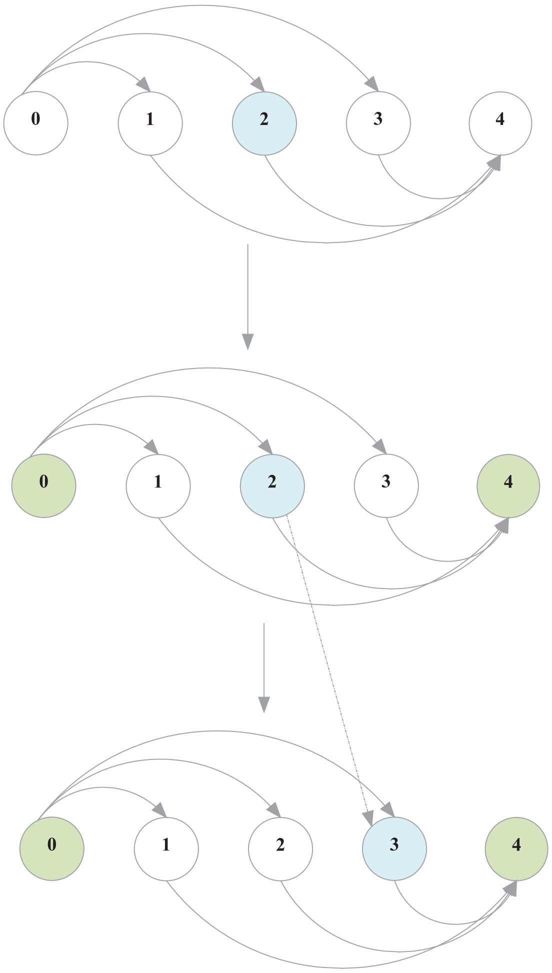

Figure 3: Scenario of mutation process

Based on the number of tasks presented in workflow, the mutation operation is performed at position, 's'. Further, the fragment search process is determined in two directions. The search process is stopped, when the present task is based on the selection of front or back task. Finally, the mutation process executes the task in a new position in detection range.

```
    Begin
    Initialize no. of tasks → $A$
    $\rho_1$ $and$ $\rho_1$ ← two chromosomes
    Declare ‘mp’ is the mutation point
    Declare $Initial-point = s,\ Final-point = mp$
    While $Initial-point \geq 0 \leq \&\& \leq task\,allocation[s] \notin front(task\,allocation[mp])do$
$Initial-point = \ Initial-point - 1$
    While $Final-point \leq A \leq task\,allocation[Final-point]\&\&back(task\,allocation[mp])do$
$Final-point = Final-point + 1$
    Move $task\ allocation$ to any position in the interval [s, mp]
    End
```

4.5 Cost-Effective Optimal Task Scheduling Model (CEOTS) Workflow

The current section presents the complete work flow of the proposed Cost-Effective Optimal Task Scheduling Model (CEOTS) that performs effectively in hybrid cloud framework. A complete work flow of the proposed model is shown in Fig. 4.

The operations of the proposed model shown in Fig. 4 are explained clearly in the following steps.

i) The consumer needs are obtained and the tasks are divided by CSP.
ii) The function of task manager is to analyse the task attributes and the factors include task_type, scheduling pattern and so on.
iii) Task information is transmitted to protocol analyzer.
iv) CEOTS model is incorporated for task scheduling in an optimal manner.
v) The arithmetic computations are carried out in CEOTS and are provided for task scheduling.

The task scheduler ensures that the resources are allocated for efficient task accomplishment based on the requirements of a user. By utilizing the computations of the proposed model, efficient resources are allocated to appropriate tasks. The process can reduce the waiting time for completion of the tasks, thereby the make span gets effectively reduced.

5 Experimental Evaluations and Discussions

The proposed model was implemented in CloudSim platform for model evaluation. Few factors such as task completion time, make span and processing cost were used to test the efficacy of the model. In order to establish that the model is optimal, the results were compared with results attained from existing models such as Resource Allocation System (RAS), Deadline Early Tree (DET) and Heterogeneous Earliest Finish Time (HEFT). Moreover, the evaluations were conducted based on two cases as given herewith.

Case 1:

This evaluation had a total of five tasks and three virtual machines were considered. The task set is as follows, $Tk = \{tk_1, tk_2, tk_3, tk_4, tk_5\}$ whereas the set of virtual machines are given as follows, $VM = \{vm_1, vm_2,\ vm_3\}$.

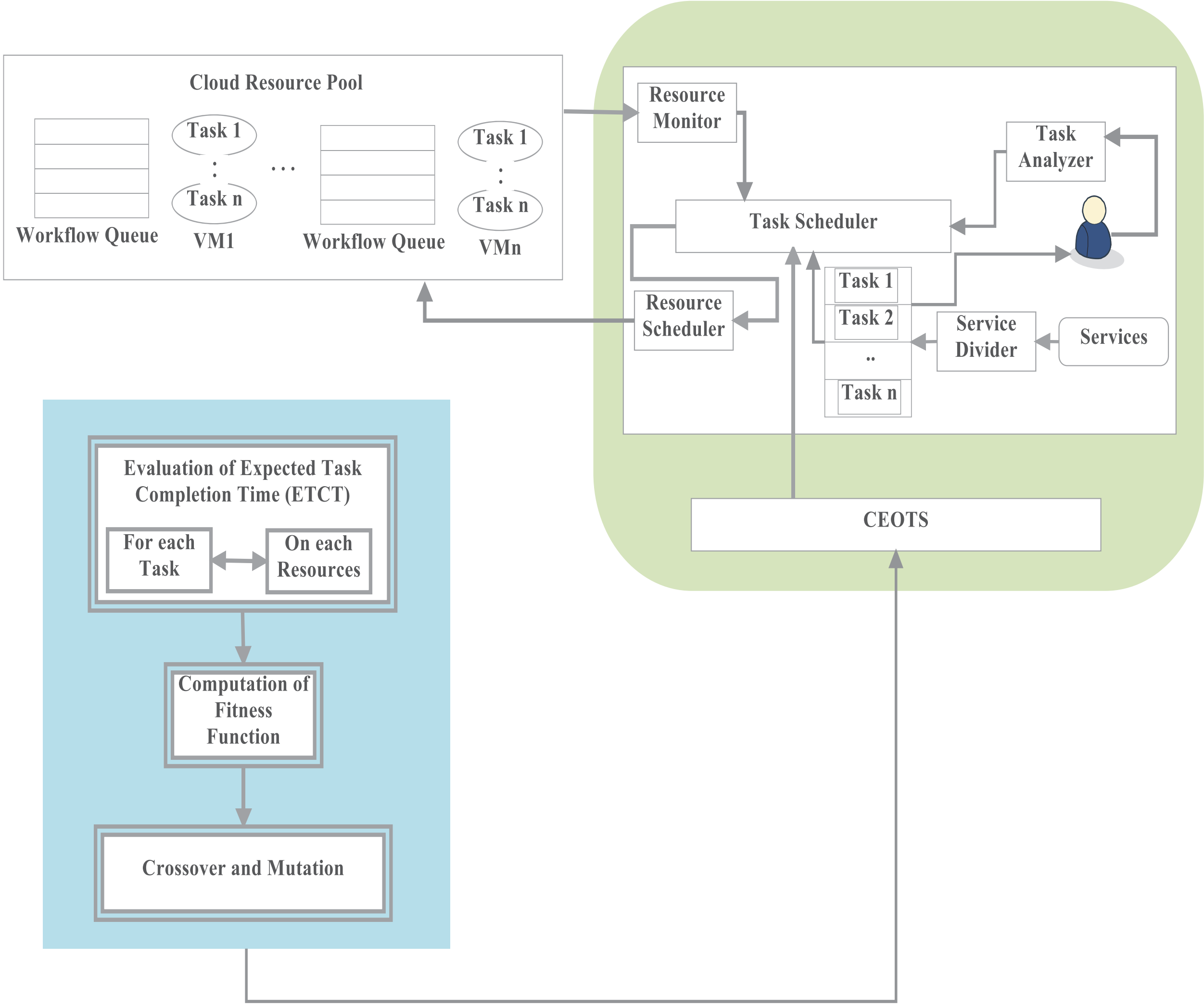

Figure 4: Workflow of Cost-Effective Optimal Task Scheduling model (CEOTS)

Case 2:

In this scenario, the task rates were considered as 50, 100, 150, 200 and 250 and considerable virtual resources from hybrid cloud are required to process the tasks. Here, the VMs were assumed to vary in the range of 20 and 40 to complete the allocated jobs. The deadline for each task is computed as follows,

$$deadline = c* \min(EPT), \quad k \geq 1 \tag{6}$$

where, 'c' denotes the deadline constraint parameter that varies based on processing.

Fig. 5 is a depiction of results achieved from comparative analysis of total processing time attained by RAS, DET, HEFT and the proposed CEOTS. It is clear from the graph that the proposed model consumed only minimal time for processing the tasks than the compared models.

The calculations to determine the ETCT, for all the tasks received from users and corresponding rates, are provided in the graph for analysis. Fig. 6 shows the results for the analysis conducted with task sets such as $Tk = \{tk_1, tk_2, tk_3, tk_4, tk_5\}$. The corresponding results are presented in Tab. 1. Based on the observations, it is clear that the proposed model consumed minimal time for processing.

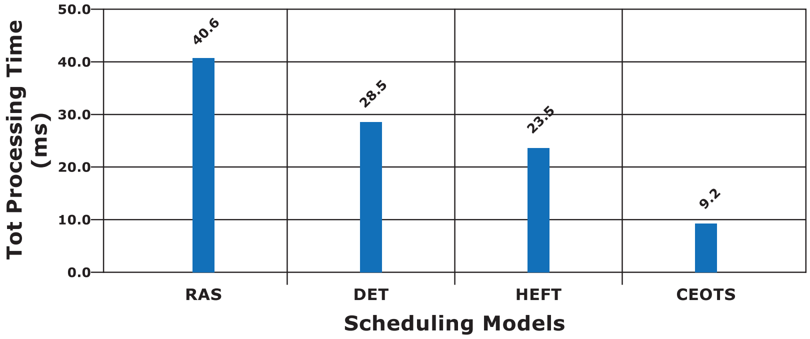

Figure 5: Comparison of total processing time in case 1

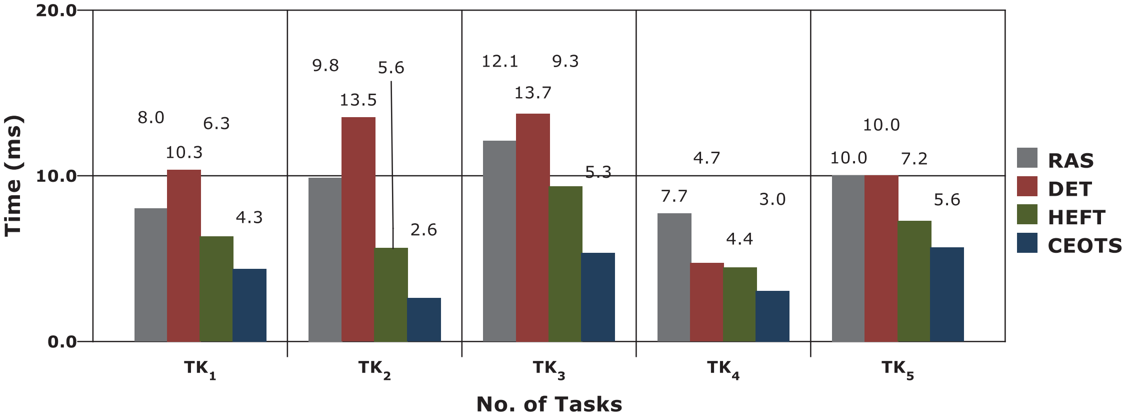

Figure 6: Execution time taken for individual tasks

Table 1: Results obtained from execution time analysis

Models	TK1	TK2	TK3	TK4	TK5
RAS	8.0	9.8	12.1	7.7	10.0
DET	10.3	13.5	13.7	4.7	10.0
HEFT	6.3	5.6	9.3	4.4	7.2
CEOTS	4.3	2.6	5.3	3.0	5.6

Fig. 7 shows the results achieved for processing time based on the virtual machine considerations, assumed in case 2. The obtained results are presented in Tab. 2. According to the results and corresponding graph, the proposed model is efficient than the compared models.

Make span is an important factor to measure the efficiency of cloud mode. It is defined as the time needed by a model to complete the tasks allotted. The model that achieves minimal make span rate infers that the scheduling model is highly efficient in resource allocation. Figs. 8 and 9 show the results achieved for case 1 and case 2 considerations, respectively. Further, the values of the respective results are provided in Tabs. 3 and 4. For case 2, the number of virtual machines varied between 20 and 40.

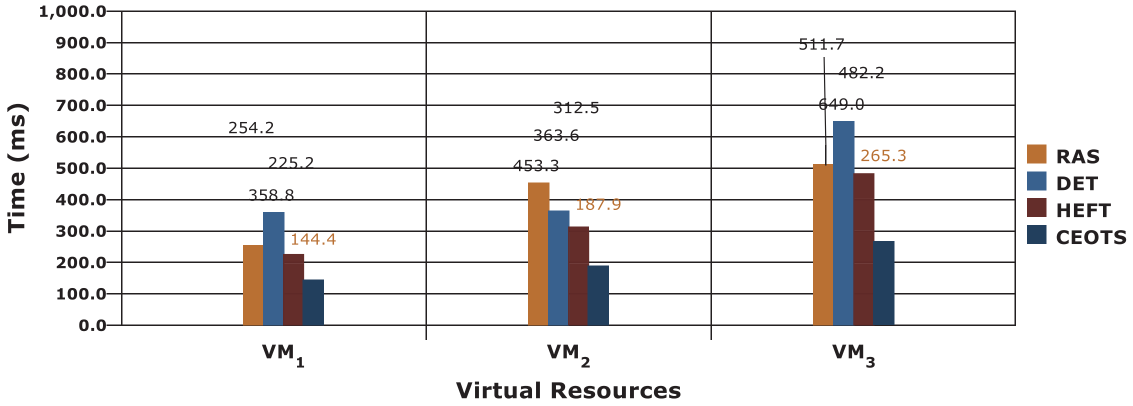

Figure 7: Comparison of total processing time based on VMS

Table 2: Results obtained from execution time analysis based on virtual machines

Models	VM1	VM2	VM3
RAS	254.2	453.3	511.7
DET	358.8	363.6	649
HEFT	225.2	312.5	482.2
CEOTS	144.4	187.9	265.3

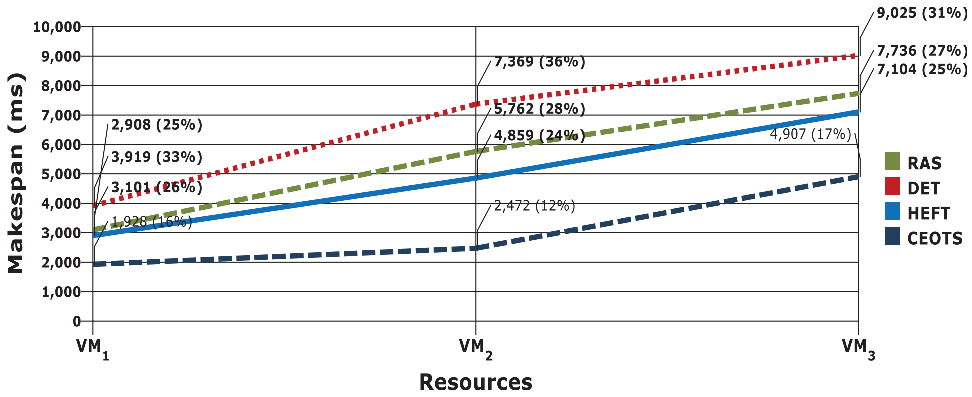

Figure 8: Make span analysis and comparison with other models

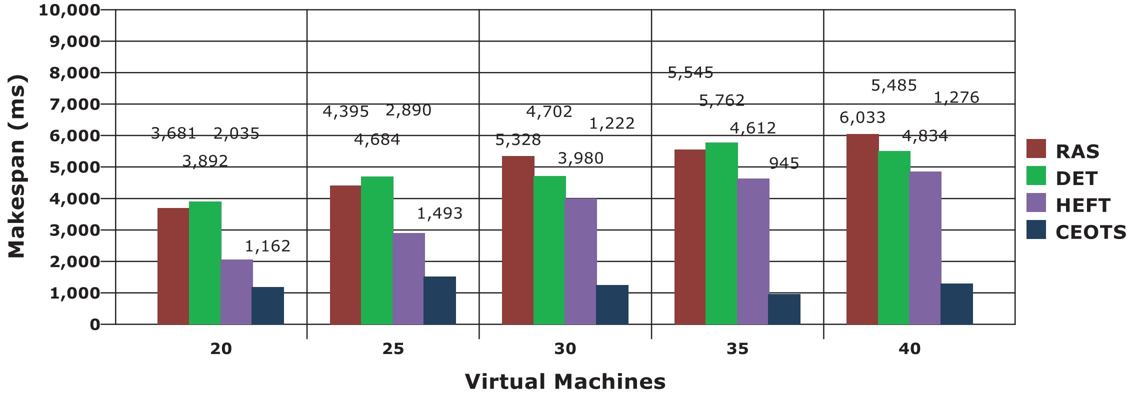

Figure 9: Make span evaluation among the models with different virtual machines

Table 3: Results obtained from make span analysis

Models	VM1	VM2	VM3
RAS	3,101	5,762	7,736
DET	3,919	7,369	9,025
HEFT	2,908	4,859	7,104
CEOTS	1,928	2,472	4,907

Table 4: Results obtained from make span analysis for number of VMS

Models	20	25	30	35	40
RAS	3,681	4,395	5,328	5,545	6,033
DET	3,892	4,684	4,702	5,762	5,485
HEFT	2,035	2,890	3,980	4,612	4,834
CEOTS	1,162	1,493	1,222	945	1,276

From the graphs, it is explicit that the proposed model achieved optimal resource scheduling for cloud resources to perform the tasks efficiently. Moreover, the results confirmed that CEOTS model utilized minimal cloud resources at a lesser make span than other models and this way the proposed model increases the system performance.

Cost effectiveness is another significant factor to evaluate the resource scheduling model in cloud. Fig. 10 shows the results of comparatively analysis in terms of overall processing costs for task executions. The obtained values are tabulated in Tab. 5. Through efficient incorporation of genetic algorithm with time computations, the proposed model used minimal resources and reduced the processing time. Hence, the processing cost to perform the tasks got reduced by the proposed model than the compared models.

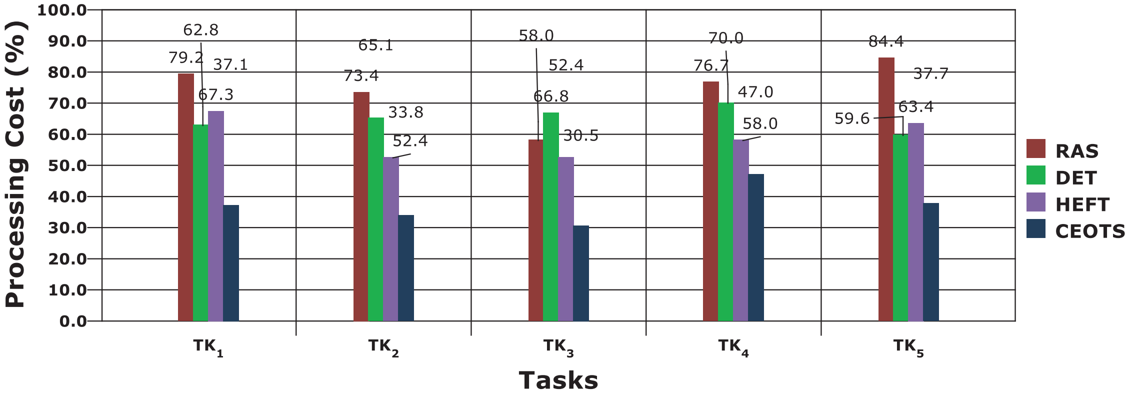

Figure 10: Processing cost *vs*. tasks

Table 5: Results obtained from cost effectiveness analysis

Models	TK1	TK2	TK3	TK4	TK5
RAS	79.2	73.4	58.0	76.7	84.4
DET	62.8	65.1	66.8	70.0	59.6
HEFT	67.3	52.4	52.4	58.0	63.4
CEOTS	37.1	33.8	30.5	47.0	37.7

6 Conclusion and Future Work

In order to resolve the task scheduling problems in cloud, the current research work proposed and developed a new model called Cost-Effective Optimal Task Scheduling Model (CEOTS). The model considered task priority, resource allocation, and the type of VM for optimal task scheduling and resource allocation. In order to reduce the cost and makespan effectively, the model performed optimal resource allocation using GA model with crossover and mutation operations. In order to enhance the results, the model used processing time computation for fitness function derivation. The proposed model was evaluated based on few factors such as make span, cost, and processing time. From the graphs and comparisons discussed in previous section, it is concluded that the proposed model achieved better results than the existing models. In future, the model can be implemented in real-time environment while reliability-based operations can also be included in the model for efficient task scheduling in hybrid cloud model.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] E. Vishnupriya and T. Jayasankar, “Performance analysis of optimum routing protocols in wireless sensor networks,” *International Journal of Innovative Science, Engineering & Technology*, vol. 2, no. 2, pp. 520–528, 2015.

[2] P. D. Saronrex and R. K. Maheswari, “Dynamic consolidation of virtual machines in cloud data centres for managing overloaded hosts under quality of service constraints,” *International Journal of Engineering Research and Development*, vol. 10, no. 3, pp. 40–47, 2014.

[3] V. R. Blessie, A. Stanislas and L. Arockiam, “A state of art scheduling algorithms in cloud environment,” *International Journal of Advanced Research in Computer Science & Technology*, vol. 2, no. 2, pp. 33–47, 2014.

[4] C. G. Hwang, C. P. Yoon and D. Lee, “Exchange of data for big data in hybrid cloud environment,” *International Journal of Software Engineering and its Applications*, vol. 9, no. 4, pp. 67–72, 2015.

[5] K. Vinoth Kumar, T. Jayasankar, V. Eswaramoorthy and V. Nivedhitha, “SDARP: Security based data aware routing protocol for ad hoc sensor networks,” *International Journal of Intelligent Networks*, vol. 1, no. 12, pp. 36–42, 2020.

[6] M. Buvana, K. Muthumayil, M. Rajinikannan and T. Jayasankar, “Optimize cryptography algorithm for efficient data security on cloud computing,” *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 1, pp. 465– 470, 2021.

[7] N. R. Mohan and E. B. Raj, “Resource allocation techniques in cloud computing–research challenges for applications,” in *Fourth Int. IEEE Conf. on Computational Intelligence and Communication Networks*, Mathura, India, pp. 556–560, 2012.

[8] S. Sharma and D. Parihar, “A review of resource allocation in cloud computing,” *International Journal of Advance research, Ideas, and Innovations in Technology*, vol. 1, no. 3, pp. 336–379, 2014.

[9] B. H. Bhavani and H. S. Guruprasad, “A comparative study of resource allocation policies in a cloud computing environment,” *Soft Computing*, vol. 3, no. 6, pp. 124–136, 2014.

[10] S. D. Patil and S. C. Mehrotra, “Resource allocation and scheduling in the cloud,” *International Journal of Emerging Trends & Technology in Computer Science*, vol. 1, no. 1, pp. 47–52, 2012.

[11] A. B. Alam, M. Zulkernine and A. Haque, “A reliability-based resource allocation approach for cloud computing,” in *IEEE 7th Int. Symp. on Cloud and Service Computing (SC2)*, Kanazawa, Japan, pp. 249–252, 2017.

[12] A. Khanna, “RAS: A novel approach for dynamic resource allocation,” in *2015 1st Int. Conf. on Next Generation Computing Technologies (NGCT)*, Dehradun, India, pp. 25–29, 2015.

[13] Y. Yuan, X. Li, Q. Wang and X. Zhu, “Deadline division-based heuristic for cost optimization in workflow scheduling,” *Information Sciences*, vol. 179, no. 15, pp. 2562–2575, 2009.

[14] L. Zeng, B. Veeravalli and X. Li, “Scalestar: Budget conscious scheduling precedence-con- strained many-task workflow applications in cloud,” *IEEE 26th Int. Conf. on Advanced Information Networking and Applications*, Fukuoka, Japan, pp. 534–541, 2012.

[15] Y. Wang and W. Shi, “Budget-driven scheduling algorithms for batches of map reduce jobs in heterogeneous clouds,” *IEEE Transactions on Cloud Computing*, vol. 2, no. 3, pp. 306–319, 2014.

[16] B. Lin, W. Guo, G. Chen, N. Xiong and R. Li, “Cost-driven scheduling for deadline-con- strained workflow on multi-clouds,” in *IEEE Int. Parallel and Distributed Processing Symp. Workshop*, Hyderabad, India, pp. 1191–1198, 2015.

[17] X. Zhou, G. Zhang, J. Sun, J. Zhou and T. Wei, “Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based HEFT,” *Future Generation Computer System*, vol. 93, no. 3, pp. 278–289, 2019.

[18] J. J. Durillo and R. Prodan, “Multi-objective workflow scheduling in amazon EC2,” *Cluster Computing*, vol. 17, no. 2, pp. 169–189, 2014.

[19] A. Verma and S. Kaushal, “A hybrid multi-objective particle swarm optimization for scientific workflow scheduling,” *Parallel Computing*, vol. 62, no. 2, pp. 1–19, 2017.

[20] C. Vecchiola, R. Calheiros, D. Karunamoorthy and R. Buyya, “Deadline-driven provisioning of resources for scientific applications in hybrid clouds with Aneka,” *Future Generation Computer System*, vol. 28, no. 1, pp. 58–65, 2012.

[21] F. Liu, B. Luo and Y. Niu, “Cost-effective service provisioning for hybrid cloud applications,” *Mobile Network Applications*, vol. 22, no. 2, pp. 153–160, 2017.

[22] M. Malawski, K. Figiela and J. Nabrzyski, “Cost minimization for computational applications on hybrid cloud infrastructures,” *Future Generation Computer System*, vol. 29, no. 7, pp. 1786–1794, 2013.

[23] X. Zuo, G. Zhang and W. Tan, “Self-adaptive learning PSO-based deadline constrained task scheduling for hybrid IAAS cloud,” *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 2, pp. 564–573, 2014.

[24] L. Bittencourt and E. Madeira, “HCOC: A cost optimization algorithm for workflow scheduling in hybrid clouds,” *Journal of Internet Services and Applications*, vol. 2, no. 8, pp. 207–227, 2011.

[25] M. Zhu, Q. Wu and Y. Zhao, “A cost-effective scheduling algorithm for scientific work- flows in clouds,” *IEEE 31st International Performance Computing and Communications Conference*, vol. 326, no. 2, pp. 256–265, 2013.

[26] L. Mo, A. Kritikakou and O. Sentieys, “Energy-QoS-time optimized task mapping on DVF- S-enabled multicores,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 11, pp. 2428–2439, 2018.

[27] R. Dhanapal, T. Akila, S. S. Hussain and D. Mavaluru, “A cost-aware method for tasks allocation on the internet of things by grouping the submitted tasks,” *Journal of Internet Technology*, vol. 20, no. 7, pp. 2055–2062, 2019.