DOI: 10.15849/IJASCA.220328.11

Int. J. Advance Soft Compu. Appl, Vol. 14, No. 1, March 2022
Print ISSN: 2710-1274, Online ISSN: 2074-8523

Efficient Task Scheduling of Virtual Machines using Novel Spectral Partitioning and Differential Evaluation Algorithm

Anwar R Shaheen, Sundar Santhosh kumar

Research Scholar, Department of Computer Science, Alagappa University, Karaikudi, India.
E-mail: shaheenanwarr14@gmail.com
Assistant Professor, Department of Computer Science, Alagappa University, Karaikudi, India.

Abstract

Task-scheduling is a major challenge in cloud computing environment that degrades the performance of the system. To enhance the performance of the system, an effective task-scheduling algorithm is needed. Hence an effective task-partitioning and task-scheduling algorithm is introduced for enhancing the system performance. To create resources (datacentre, broker, Virtual Machine - VM and cloudlet) in a dynamic way through the use of CloudSim. In addition, this study intended to perform task-partitioning and task-scheduling in an effective manner by utilizing the novel spectral partitioning - (SP) and differential evaluation algorithm - (DEA). At first, the task and datacentre is initialized. Subsequently, task-partitioning is performed using the novel SP. It includes a series of steps in which a Laplacian matrix is computed initially. Then based on the Eigen-values and Eigen-vectors of the Laplacian matrix the tasks are partitioned. Followed by this, task-scheduling is performed with the employment of proposed novel DEA. The process comprise the following series of steps such as threshold calculation, mutation, crossover, selection and knee solution for achieving efficient task-partitioning and scheduling. The performance of the proposed system is evaluated by comparing it with other traditional methods. And validated in terms of service cost, load balancing, makespan and energy consumption. The results proved the efficacy of the introduced system. The overall results obtained from comparative analysis also reveal that proposed method outperformed other traditional techniques thereby accomplishing effective task scheduling of VMs in cloud computing environment.

Keywords: *Cloud computing environment, Virtual Machines, Task Scheduling, Novel Spectral Partitioning and Differential Evaluation Algorithm.*

Received 24 November 2021; Accepted 23 February 2022

1 Introduction

Cloud computing is current requisition for resource availability of computer system, particularly computing power and data storage in the without active user management directly. In addition, a VM – Virtual Machine acts similar to a real-computer. It can also be said as a computer within the same computer. Task scheduling is vital and significant area in cloud environment. Task-scheduling is performed on VMs in cloud-computing environment to select appropriate resources for executing the task by considering few metrics and limitations. This task scheduling enhances the effective resource utilization and produces minimum response time. Thus the implementation of submitted tasks occurs within a probable minimum time. This paper (Pradeep & Jacob, 2018) introduced a hybrid CHSA (Cuckoo Harmony Search Algorithm) to enhance the optimization. The algorithms are efficiently integrated to perform process-scheduling. In accordance with this, an innovative multi-objective function has been introduced by associating energy consumption, cost, credit, message usage and penalty. Thus the proposed hybrid algorithm' performance has been compared with various methodologies such as current hybrid CGSA (Cuckoo Gravitational Search Algorithm), distinct HS and CS methodology with several multi-objective metrics. The proposed method has been analysed to validate its efficacy. The results exhibited that the proposed method minimized cost, energy consumption, memory usage, penalty thereby maximizing credit when compared to traditional methods.

In addition, this study (Jena, 2017) focussed on effective task-scheduling by utilizing ABC (Artificial Bee Colony method). The recommended technique optimized the cost, energy, processing time, resource utilization in cloud environment. The outcomes attained by task scheduling ABC is simulated via CloudSim. The recommended methodology afforded an ideal stable outcomes for multi-objectives. The outcomes are comparable to traditional scheduling methodologies. On contrary, this article (Sreenu & Malempati, 2019) explored an optimization algorithm named as MFGMTS (Modified Fractional Grey-wolf optimizer for Multi-objective Task Scheduling in cloud environment. The recommended strategy afforded minimum execution cost, communication time, execution time, energy consumption as well as resource utilization. Additionally, this paper (Arunarani, Manjula, & Sugumaran, 2019) explored a survey for task scheduling methodologies as well as related parameters appropriate for cloud environments. Moreover, this study proposed an innovative systematic methodology that takes into account the interactions as well as security demands for effective task scheduling. The empirical outcomes exhibited that the proposed technique can efficiently minimize security risk (Abazari, Analoui, Takabi, & Fu, 2019).

Awareness about the availability of VMs (Virtual Machines) are the main task-scheduling process in cloud environment. Nevertheless, uncertainty and dynamic alterations in the availability of VMs. Hence, the quality requirements corresponding to task services could not be fulfilled which has a serious impact on

task scheduling ability of cloud environment. Though conventional methods attempted to perform task-scheduling, they have been ineffective with respect to service cost, load balancing, makespan and energy consumption. To solve these issues, the present work proposed a novel Spectral Partitioning and Differential Evaluation Algorithm to efficiently partition the tasks and perform effective task scheduling on VM (Virtual Machines) in cloud computing environment.

The proposed DEA is capable of minimizing the completion time, enhancing the degree of load balancing and reducing energy consumption. To improvise global search capability in initial phase and local search capability in final phase, adaptive-crossover factor improving approach and adaptive-zooming factor-mutation approach are adopted. Concurrently, the selection approach are strengthened to maintain the population diversity in final phase. In addition, the proposed SP are the generally capable of showing effective outcomes that works on the finite-element meshes and planar graphs of bounded degree. It can also be utilized in VLSI simulation and several empirical researches have revealed that, spectral methodologies are capable of determining good matrices and graph partition. Due to these advantages, this study proposes SP and DEA for scheduling the tasks of VMs in an effective way.

The major contributions involved in this study are listed below

- To dynamically create resources (datacentre, broker, VM and cloudlet) through the use of CloudSim to perform task-scheduling.
- To effectively perform task-partitioning via the use of novel spectral partitioning algorithm.
- To perform task-scheduling in an efficient way through the proposed novel Differential Evaluation Algorithm.
- To evaluate the performance of the proposed methodology by comparing it with other conventional methods.

1.1 Paper Organization

The basic ideas about task scheduling on VMs in cloud environment is presented in the above section I. The upcoming section II describes the various existing works related to efficient task scheduling in cloud computing environments. The proposed methodology for effective task scheduling in cloud environment is explained in section III. The results obtained from the proposed methodology is given in section IV. The overall proposed system for effective task scheduling is summarized in section V.

2 Review of Existing Work

The development of security aware IC (Industrial Control) afforded an approach for scheduling based on security in industrial applications of cloud environment.

This study (Meng et al., 2020) introduced a dynamic scheduling technique based on security for resource allocation in real-time in Industrial Control Systems (ICS). Initially, a three level (3L) security model has been designed for cloud resources as well as tasks in ICS. In addition, a heterogeneous two-tier cloud architecture has been introduced. Consequently, a scheduling technique based on security using PSO – Particle Swarm Optimization has been explored to allocate resources by considering the security issues. To handle with the edge resource dynamics as well as mobility corresponding to industrial applications, this study introduced a dynamic scheduling methodology on the basis of dynamic workflow for optimization in real-time. Experiments has been carried out to validate the efficiency of the proposed method. The results explored that this paper accomplished a good balance amongst security and scheduling performance. In addition, a novel united methodology has been proposed in (Zheng, Wu, & Nie, 2017) that performs task scheduling as well as allocates the contents of locked cache of individual task to L1 caches (local caches), the L2 cache (two level cache). The experimental results exhibited the efficiency of the proposed methodology.

Similarly, this paper (Cao et al., 2019) address the lifetime optimization issue for MC-embedded systems through designing efficient task scheduling approaches. Simulation results explored that the proposed method can expand the lifetime of the system thereby satisfying the task timeliness and safety requirement constraints. In addition, this article (Abd Elaziz, Xiong, Jayasena, & Li, 2019) explored an alternative technique for cloud task-scheduling issue that intended to reduce make-span which needed to schedule various tasks on several VMs (Virtual Machines). The proposed technique relied on enhancement of MSA (Moth Search Algorithm) by utilizing DE (Differential Algorithm). The results attained from experiments explored that the proposed methodology performed better than other traditional methods. Moreover, this study (Mahmood, Khan, Albalooshi, & Awwad, 2017) proposed task scheduling in real-time for Dynamic Voltage Scaling (DVS) permitted multi-processor systems. Subsequently, it introduced a GA (Genetic Algorithm) which is hybridized with SEA (Stochastic Evolution Algorithm) for allocating as well as scheduling real-time chores with superior restraints. The proposed technique has been analysed comparatively with GA, CS (Cuckoo Search), PSO (Particle Swarm Optimization) and ACO (Ant Colony Optimization). Simulation has been undertaken to assess the efficiency of the proposed methodology. The results showed that the proposed technique outperformed other algorithms. On the other hand, this paper (Ghoshdastidar & Dukkipati, 2017) explored a PPM- Planted-Partition Model for SRN-HG (Sparse Random Non-uniform-Hyper Graphs) which simplifies the SBM-Stochastic Block Model. The proposed system showed effective results on analysis.

On contrary, this article (Midya, Roy, Majumder, & Phadikar, 2018) introduced a three-tier design comprising of centralized cloud, vehicular cloud and road-side cloudlet. This study also proposed optimized task scheduling and resource allocation methodologies to effectively assist various count of task requests received from road-users while managing enhanced QoS (Quality of Service). The optimization process is undertaken using the introduced HAPSO (Hybrid

Adaptive Particle Swarm Optimization) that is the combination of GA and APSO. Simulation is undertaken using SUMO. Thus the results exhibited that HAPSO converges quicker than standard PSO and S-APSO (Self-Adaptive Particle Swarm Optimization). Similarly, this study (Hu, 2020) addressed the issue of profit maximization of user-tasks in real-time by resource allocation and task scheduling in C-RANs (Cloud-Radio Access Networks). Subsequently, MPA (Maximum Profit Algorithm) is introduced to attain the solution of Integer Linear Programming – ILP by PRT (Profit based Rounding Technology). The results obtained from simulations explored that the proposed MPA is easy to execute in C-RAN. Likewise, the potent rewards and challenges of incorporating CPSs (Cyber Physical Systems) and MCSs (Mixed Criticality Systems) have been introduced in (Capota, Stangaciu, Micea, & Curiac, 2019).

Though there exists no scheduling method that is appropriate for every application fields of cyber-physical systems, few methodologies are highly suitable considering definite attributes. Accordingly, this study (Konar, Bhattacharyya, Sharma, Sharma, & Pradhan, 2017) proposed a hybrid GA based strategies for task scheduling (real-time) in multi-processor environment. The experiments have been performed to validate the efficacy of the proposed method which revealed effective outcomes. Additionally, this article (Liu et al., 2019) recognizes and studies an innovative privacy or security problem for auto-mobile vehicles. Simulation has been conducted to determine its efficacy. In the same way, initial semi-partitioned scheduling methodology has been introduced in (Hobbs, Tong, Bakita, & Anderson, 2021) which is ideal for Soft Real Time (SRT) sporadic task schemes which can deal with the dynamic workload alterations. Similarly, (Gawali & Shinde, 2018) introduced a heuristic scheme which associates MAHP (Modified Analytic Hierarchy Process), LEPT (Longest Expected Processing Time), BATS integrated with BAR as well as dive and conquer techniques to perform resource allocation and task scheduling. But the response time and turn-around time has to be enhanced. It is a drawback in this study. In addition, this paper (Taheri, Khonsari, Entezari-Maleki, & Sousa, 2020) developed a two-phase H-TSA (Hybrid-Task Scheduling Algorithm) on the basis of task graph decomposition by employing spectral partitioning. Results showed the efficacy of the proposed method. On contrary, this article (Y. Yang et al., 2018) introduced a framework to study the stability amongst energy consumption and service delay. Experiments has been performed to evaluate the performance of the proposed methodology. The results showed that the proposed framework can accomplish gains in the performance (delay-energy) under varied service and network conditions. This study has to be extended further in highly complex H-FN (Heterogeneous-Fog Networks).

In addition, this study (Zhang, Hu, He, Liu, & Chen, 2019) examined the expandability of two TSA such as linear programming and dynamic programming algorithms in real-time. It specifically concentrate on their stability and operating efficacy. The findings explored that linear programming algorithm performed better than dynamic programming algorithm with respect to expandability. Similarly, this article (Srichandan, Kumar, & Bibhudatta, 2018) explored a task scheduling methodology via the use of hybrid scheme that combined preferred

features of widely utilized BF (Bacterial Foraging) and GA in cloud computing. Experiments were performed and the results explored that the proposed system outperformed other traditional techniques. Additionally, SGO (Social Group Optimization) and SJF (Shortest Job First) algorithms has been introduced in (Praveen, Rao, & Janakiramaiah, 2018) for resource allocation, to reduce the make-span-time as well as enhance throughput. The results showed the efficacy of the introduced technique in terms of throughput and make-span-time. On contrary, a TSA on the basis of ACO (Ant Colony Optimization) has been proposed in (Moon, Yu, Gil, & Lim, 2017) that performs task allocation for cloud users to VMs in cloud environments. The introduced method resolved the global-optimization issue thereby proving its effectiveness.

Likewise, a TSA named W-Scheduler has been introduced in (Sreenu & Sreelatha, 2019) that rely on WOA (Whale Optimization Algorithm) and multi-objective model. This study examined the proposed method's performance by comparing it with the existing techniques. The results obtained from experiments conclude that the introduced system can perform task scheduling to VMs efficiently. In addition, this study (Xavier & Annadurai, 2019) concentrated on reducing the overall make-span time with efficient LB (Load Balancing) through modelling of swarm intelligence corresponding to social spider based on random selection. Simulation has been carried out. The performance of the proposed system is evaluated via the use of CloudSim toolkit. A comparative analysis has been performed with other existing methodologies. The validation outcomes exhibited that there exists enhancement in reducing the make-span with stable task-distribution. In the same way, this article (Beegom & Rajasree, 2019) introduced a discrete version of PSO such as Integer-PSO for task scheduling in cloud environment. Empirical studies on various kinds of task set characterising burst traffic and normal traffic in cloud environment exhibited that the proposed scheme is better. Additionally, a LOA (Lion Optimization Algorithm) has been proposed in (Almezeini & Hafez, 2017) for cloud computing. The recommended TSA is compared with GA and PSO. The results from comparative analysis revealed that the proposed method performed better than traditional methods. On contrary, this paper (Soltani, Soleimani, & Barekatain, 2017) reviewed TSAs. The study also classified as well as compared necessary metrics for scheduling. Additionally, this article (Aziza & Krichen, 2018) proposed a genetic methodology to model and optimize task scheduling issue in cloud computing. The empirical outcomes explored that the proposed solution competes with the existing task-scheduling methods. For this purpose, this study introduced a decision-support method that rely on the CloudSim. The attained results exhibited that the proposed method outperformed existing techniques with respect to cost-processing. The attained outcomes minimized the make-span time when compared to other traditional methods.

3 Proposed Methodology

The study intended to perform task scheduling of Virtual Machines in an effective way which is challenging task in the cloud computing environment. Though

conventional systems attempted to perform effective task scheduling, they lacked in terms of service cost and load balancing. Energy consumption and makespan has also been found to be more for the existing methods. To resolve this, the proposed methodology performed several tasks as shown in figure.1.

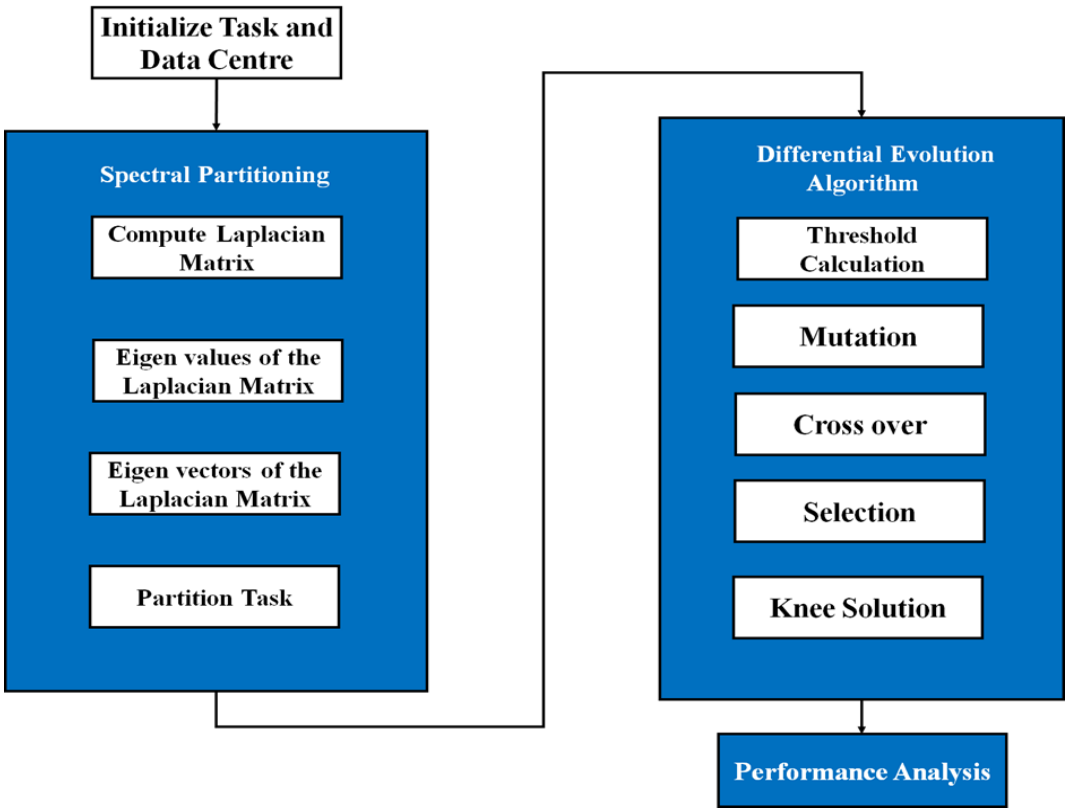

Figure.1. Overall view of the proposed method

In figure.1, At first, the task and datacentre are initialized. Then the tasks are partitioned using novel SP-Spectral Partitioning algorithm. The spectral partitioning method includes a series of steps where the Laplacian matrix is computed. On the basis of Eigen vectors and Eigen values of the Laplacian matrix, the tasks are partitioned. Task-scheduling is performed using the novel DEA – Differential Evaluation Algorithm. In DEA, the threshold is initially calculated followed by a series of process such as mutation, crossover, selection and knee solution. Finally, performance analysis is carried out to evaluate the efficiency of the proposed methodology. For example, when multiple tasks possess the same priority, RR (Round Robin) scheduling will be generally utilized to schedule them which does not afford special-priority to important tasks and also minimize the comprehension. However, when a high priority task happens, those tasks must be given first priority and implemented. To achieve this, the present study proposes Spectral Partitioning (SP) and Differential Evaluation Algorithm (DEA).

3.1 Novel Spectral Partitioning (SP) Algorithm

The spectral document partitioning evolved in recent years as wide-spread partitioning method that motivated the evolving standard functions as well as the developing methodology to generate further proper clusters. It utilizes the Eigen vectors corresponding to graph matrices that is received from documents. This methodology relies on the undirected graph (weighted) concept. It represents the document collection that is document corpus. Idea behind proposed SP is that, it determines the splitting values that partition graph vertices into a two sets with the field-vector more than splitting values and field-vector less than splitting values which is termed as fielder cut. The SP methodologies utilize the fieldler-vector for determining the graph's small-separator. This methodology comprises of the below steps which is shown in table-1.

Table 1: Algorithm I

Algorithm I: Novel SP
Input – Similarity Matrix $T \in Q^{n*n}$ and the count of partition A. Let E be the – weighted matrix **Output -** Count of partitions A **Step 1 –** Construct similarity matrix by cosine similarity which is given by $CS(e_1, e_2) = \frac{e_1 * e_2}{\lvert e_1 \rvert * \lvert e_2 \rvert}$ **Step 2 –** Calculate the Laplacian that is un-normalized by utilizing $IJ = P - E$ **Step 3 –** Evaluate the partition A with Eigen vectors $F_1, F_2, ..., F_K$ of IJ **Step 4 –** Let $F \in Q^{n*n}$ that comprises vectors $F_1, F_2, ..., F_K$ as a column **Step 5 –** Let $z_b \in Q^k$ be the vector equivalent to i^{th} row of F **Step 6 –** Group points z_b in Q^k with partitioning methodology into $\{A_1, A_2, ..., A_n\}$

3.2 Novel Differential Evaluation Algorithm (DEA)

Proposed DEA is an optimization technique utilized for real-value multi-dimensional functions that utilizes a population corresponding to individual outcomes. The technique do not need gradient information that represents that optimization issue do not require to be variant. The algorithm hunts the design space by managing a population corresponding to individual solutions as well as building innovative solutions by uniting traditional ones in accordance with a particular process. The individuals with finest objective values are placed in the succeeding iteration pertaining to the algorithm in a way which the individual's

innovative objective value is enhanced establishing a population part. If not, an innovative objective value is rejected. The process replicates itself till a specific termination condition is satisfied. The DEA algorithm is given below in table-2.

Table 2: Algorithm II

Algorithm II: Novel DEA

Information or data:
GH – size of the population, L – Mutation Factor, XY – Crossover Probability, MFE – Maximum count of Function Evaluations
INITIALIZATION $D = 0$; Initialize every GH individuals with positions in search space randomly;
While $FE < MFE$ **do**
for $p \leftarrow 1$ **to** GH **do**
GENERATE three individuals z_{b1}, z_{b1}, z_{b1} from the present population in a random way.
These should be different from one another, even from individual z_p, that is $b_1 \neq b_2 \neq b_3 \neq p$
MUTATION Utilizing the formula from donor vector given by $w_p = J_{b1} + L(J_{b2} - J_{b3})$
CROSSOVER Trial Vector T_p is established from the target vector elements J_p or the donor vector elements w_p which is given as:
$t_{p,q} = \{ \begin{matrix} w_{p,q} & \text{if } b_{p,q} \leq XY \\ z_{p,q}, & otherwise \end{matrix}$ or $q = q_{rand}$
Here $p = \{1,..,GH\}$, $q = \{1,...,E\}$ is a random number which is uniformly distributed and is generated for each q and $q_{rand} \in \{1,...,E\}$ is a integer (random) that is utilized to confirm that $T_p \neq J_p$ in every cases
EVALUATE If $f(T_{p)} \leq f(J_p)$ then swap the J_p –individual in the population possessing the trial vector T_p
$FE = FE + GH$
end
$D = D + 1$;
end

The proposed DEA is a population relying stochastic search method to solve optimization issues in continuous-space. Alike other algorithms, DEA utilizes direction and distance information from the current population for guiding its upcoming search. Basic vital idea behind it is the method to generate trial vectors based on the management of difference vector and target vector. When trail vector affords minimum objective function in comparison to member of a pre-determined population, presently produced trail-vector substitutes the vector as well as compares in subsequent generation. This algorithm initiates with random initialization of population. After the initialization, generation of mutation vectors

occur based on individual target vector or population member in present population. Finally, cross over is performed based on the above pseudo-code to accomplish effective task scheduling of VMs.

4 Results and Discussion

The proposed novel SP and DEA is analysed by comparing it with other traditional methodologies such as FOG-AMOSM and SGA. The comparison is done with respect to service cost and load balancing. It is explained below.

4.1 Comparison of the existing and proposed methodology with respect to load balancing for task scheduling

The proposed techniques are compared with existing methodologies in terms of load balancing. It is shown in the below table-3.

Table 3: Analysis with respect to load balancing (in milliseconds)

	FOG-AMOSM	SGA	PROPOSED
VM1	12	13	16
VM2	13	15	15
VM3	13	15	15
VM4	13	13	12
VM5	12	12	12
VM6	12	11	11
VM7	12	11	10
VM8	12	10	8

The graphical form of load balancing performance for task scheduling is shown in the below figure.2.

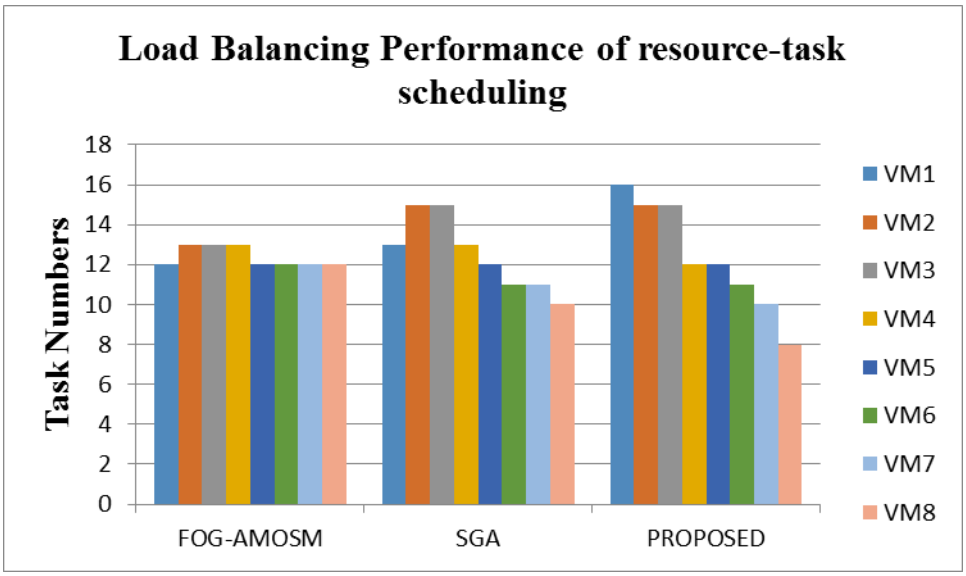

Figure.2. Comparative analysis of Load Balancing for resource-task scheduling with existing (M. Yang et al., 2020) and proposed methods

The load balancing performed by existing and proposed methods for VM1, VM2, VM3, VM4, VM5, VM6, VM7 and VM8 have been analyzed. From the results, it is clear that, load balancing rate of FOG-AMOSM for VM1 is 12ms. This rate deviates while balancing the loads of other VMs. Similarly, SGA also showed deviations in load balancing that consumed 10ms for balancing the load of VM8. In this case, time taken for load balancing by SGA decreased. However, the proposed method consumed minimum time than existing SGA and FOG-AMOSM to balance the load by taking 16ms to balance the load of VM1, while, the time minimized to 8ms for balancing the load of VM8. This steady reduction in the time for load balancing accomplished by the proposed system makes it effective than conventional system. Thus, the results showed that proposed method outperformed other methods thereby exhibiting enhanced load balancing performance to schedule the tasks.

4.2 Comparison of the existing and proposed methodology with respect to service cost for task scheduling

The proposed methods are compared with existing methodologies in terms of service cost for different number of tasks. Obtained outcomes are shown in the below table-4.

Table 4: Service Cost

TASKS	20	40	60	80	100	120	140	160
CRRSA(Rs)	90	160	230	320	400	480	580	630
SGA(Rs)	95	170	250	325	420	500	590	650
FOG-AMOSM(Rs)	85	155	220	310	390	460	560	610
PROPOSED(Rs)	70	130	200	280	350	440	520	580

The graphical form of service cost performance for task scheduling is shown in the below figure.3.

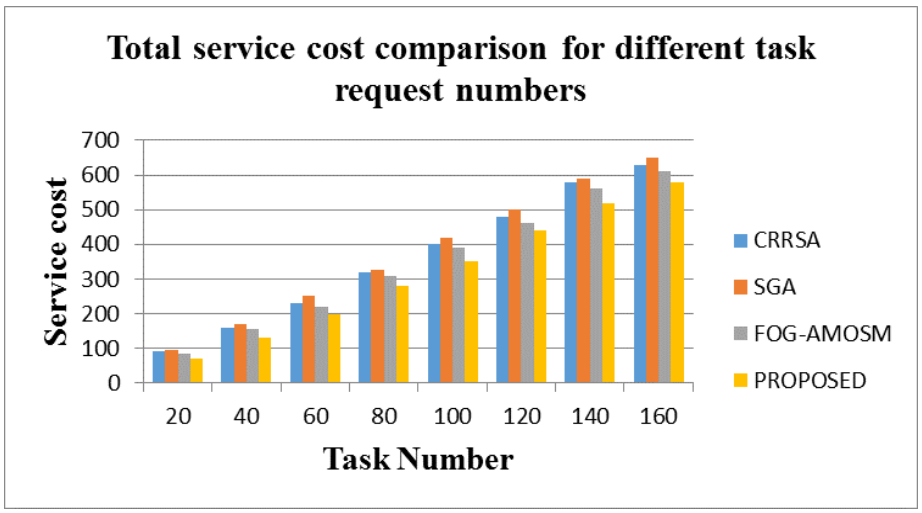

Figure.3. Comparative analysis of service cost for resource-task scheduling with existing (M. Yang et al., 2020) and proposed methods

The service cost varies based on the number of tasks. Service cost for 20 tasks is 90.Rs while using CRRSA. Concurrently, the cost is 95.Rs while using SGA, it is 85.Rs while utilizing FOG-AMOSM, whereas, the service cost is only 70.Rs for 20 tasks while using proposed method. Similarly, as the number of tasks increases, the service cost also increases. However, in comparison to traditional method, service cost is found to be less while using proposed system which shows its effectiveness. Thus, the results showed that proposed method outperformed other methods thereby minimizing the service cost to perform task-scheduling. Thus efficient task scheduling on VMs is performed through the proposed novel SP and DEA. In addition, analysis has been carried in terms of makespan, energy consumption with respect to number of tasks and energy consumption with respect to number of iterations. The obtained results are shown in figure.4, figure.5 and figure.6.

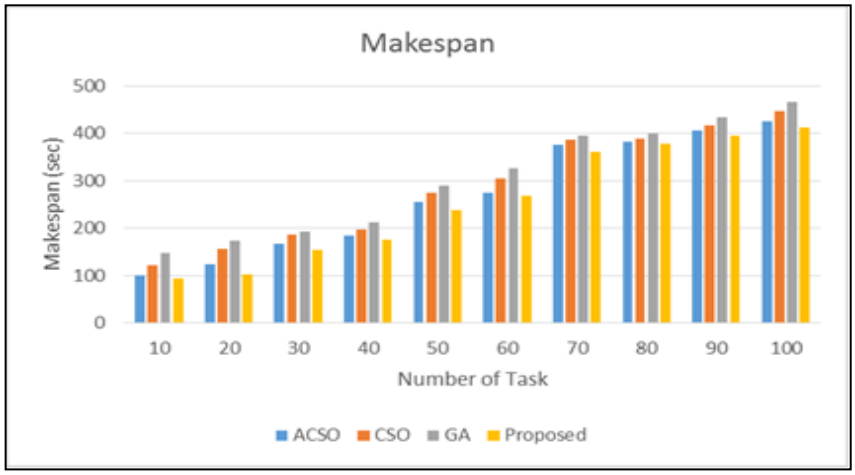

Figure.4. Comparative analysis of makespan for resource-task scheduling with existing (Balaji, Kiran, & Kumar, 2021) and proposed methods

Generally, as the number of tasks increases, the makespan also increases. In this study, tasks have been considered in the order 10, 20..100. Adaptive Cat Swarm Optimization (ACSO), Cat Swarm Optimization (CSO) and Genetic Algorithm (GA) have been the conventional methods considered for analysis. From the results is figure.4, it is clearly found that, as the tasks increase, makespan time increase while using all the methods. However, in comparison to ACSO, CSO and GA, the proposed system is found to explore minimum makespan that shows its efficacy than conventional methods with less than 400 secs when number of assigned tasks are 100.

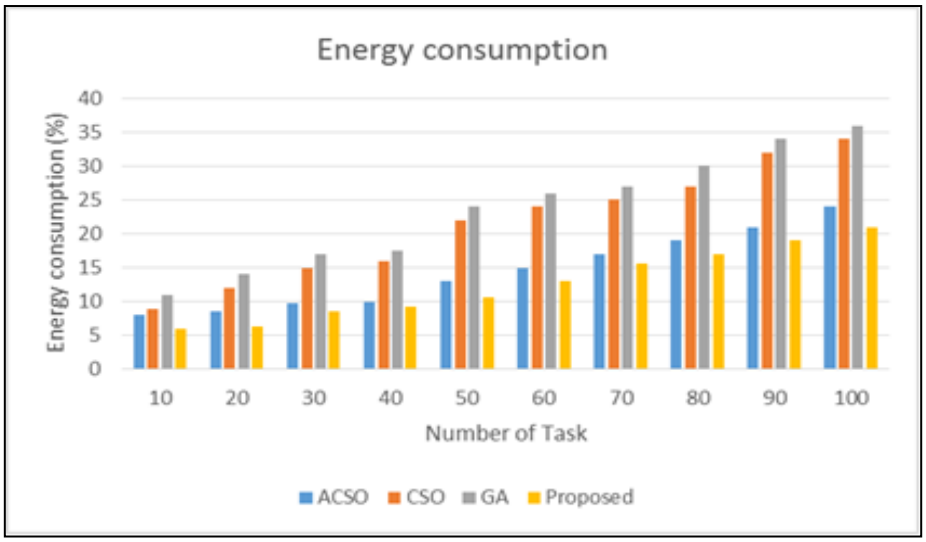

Figure.5. Comparative analysis of energy consumption with respect to number of tasks for resource-task scheduling with existing (Balaji et al., 2021) and proposed methods

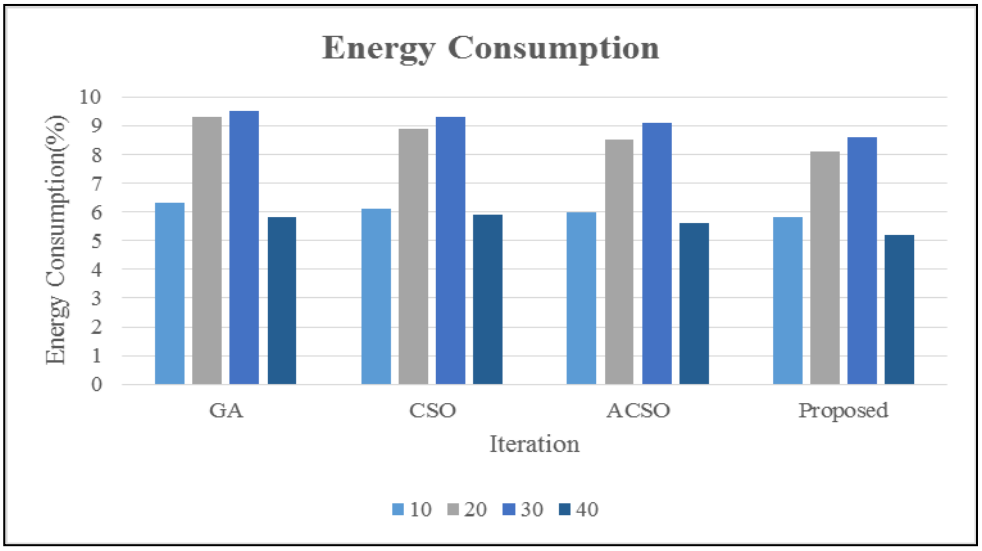

Figure.6. Comparative analysis of energy consumption with respect to iteration for resource-task scheduling with existing (Balaji et al., 2021) and proposed methods

From figure.5 and figure.6, it is found that, energy consumption of the proposed system decreases as the task and iteration increases in comparison to conventional systems. Generally, the proposed spectral partitioning is capable of affording good clustering outcomes and easy execution. On the other hand, the proposed DEA is generally able to determine the true-global minimum corresponding to the multi-modal search space irrespective of the initial-parameters. It is also capable of utilizing minimum control parameters and able to converge faster. Due to these advantages, the proposed system is found to show better outcomes than conventional methods in terms of service cost, load balancing, energy consumption and makespan for task scheduling.

5 Conclusion

Task-scheduling is important in cloud-computing environment to increase the system performance. Thus novel techniques have been introduced to partition the tasks and perform task scheduling. Novel Spectral Partitioning – SP was proposed to effectively partition the tasks and novel Differential Evaluation Algorithm - DEA is proposed to effectively perform task scheduling. The performance of the proposed system was analysed in terms of service cost, load balancing, energy consumption with respect to number of iterations, energy consumption in terms of number of tasks and makespan. Time taken for Load balancing the VMs to perform task scheduling is found to be reduced while using proposed system than conventional system at a rate of 16ms while load balancing VM1 which gets reduced to 8ms while balancing load of VM8. In addition, service cost increases based on the number of tasks. As shown in comparative analysis, service cost has been 610.Rs for scheduling 160 tasks while considering existing FOG-AMOSM. Whereas, service cost reduced to 580.Rs for scheduling the same 160 tasks while considering the proposed system. With respect to makespan, proposed system explored minimum makespan that shows its efficacy than traditional methods with less than 400 secs when number of tasks to be scheduled are 100. Moreover, the energy consumption rate of the proposed system has been low than conventional systems that has been 21% for scheduling 100 tasks. The energy consumption rate of the proposed system has been 5.1% at 40^{th} iteration. This energy consumption rates have been found to be less than traditional GA, CSO and ACSO. Thus, the results confirmed the efficacy of the proposed methodology in performing task-scheduling thereby reducing service cost and increases the load balancing capability. Thus overall system performance is found to be improved through the proposed methodologies for task-scheduling of VMs than conventional methods.

References

[1] Abazari, F., Analoui, M., Takabi, H., & Fu, S. (2019). MOWS: multi-objective workflow scheduling in cloud computing based on heuristic algorithm. *Simulation Modelling Practice and Theory, 93*, 119-132.

[2] Abd Elaziz, M., Xiong, S., Jayasena, K., & Li, L. (2019). Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution. *Knowledge-Based Systems, 169*, 39-52.

[3] Almezeini, N., & Hafez, A. (2017). Task scheduling in cloud computing using lion optimization algorithm. *algorithms, 5*(7).

[4] Arunarani, A., Manjula, D., & Sugumaran, V. (2019). Task scheduling techniques in cloud computing: A literature survey. *Future Generation Computer Systems, 91*, 407-415.

[5] Aziza, H., & Krichen, S. (2018). Bi-objective decision support system for task-scheduling based on genetic algorithm in cloud computing. *Computing, 100*(2), 65-91.

[6] Balaji, K., Kiran, P. S., & Kumar, M. S. (2021). An energy efficient load balancing on cloud computing using adaptive cat swarm optimization. *Materials Today: Proceedings*.

[7] Beegom, A. A., & Rajasree, M. (2019). Integer-pso: a discrete pso algorithm for task scheduling in cloud computing systems. *Evolutionary Intelligence, 12*(2), 227-239.

[8] Cao, K., Xu, G., Zhou, J., Chen, M., Wei, T., & Li, K. (2019). Lifetime-aware real-time task scheduling on fault-tolerant mixed-criticality embedded systems. *Future Generation Computer Systems, 100*, 165-175.

[9] Capota, E. A., Stangaciu, C. S., Micea, M. V., & Curiac, D.-I. (2019). Towards mixed criticality task scheduling in cyber physical systems: Challenges and perspectives. *Journal of systems and software, 156*, 204-216.

[10] Gawali, M. B., & Shinde, S. K. (2018). Task scheduling and resource allocation in cloud computing using a heuristic approach. *Journal of Cloud Computing, 7*(1), 1-16.

[11] Ghoshdastidar, D., & Dukkipati, A. (2017). Consistency of spectral hypergraph partitioning under planted partition model. *Annals of Statistics, 45*(1), 289-315.

[12] Hobbs, C., Tong, Z., Bakita, J., & Anderson, J. H. (2021). Statically optimal dynamic soft real-time semi-partitioned scheduling. *Real-Time Systems*, 1-44.

[13] Hu, C. C. (2020). Profit-Based Algorithm of Joint Real-Time Task Scheduling and Resource Allocation in C-RANs. *IEEE Internet of Things Journal*.

[14] Jena, R. (2017). Task scheduling in cloud environment: A multi-objective ABC framework. *Journal of Information and Optimization Sciences, 38*(1), 1-19.

[15] Konar, D., Bhattacharyya, S., Sharma, K., Sharma, S., & Pradhan, S. R. (2017). An improved Hybrid Quantum-Inspired Genetic Algorithm (HQIGA) for scheduling of real-time task in multiprocessor system. *Applied Soft Computing, 53*, 296-307.

[16] Liu, S., Guan, N., Ji, D., Liu, W., Liu, X., & Yi, W. (2019). Leaking your engine speed by spectrum analysis of real-Time scheduling sequences. *Journal of Systems Architecture, 97*, 455-466.

[17] Mahmood, A., Khan, S. A., Albalooshi, F., & Awwad, N. (2017). Energy-aware real-time task scheduling in multiprocessor systems using a hybrid genetic algorithm. *Electronics, 6*(2), 40.

[18] Meng, S., Huang, W., Yin, X., Khosravi, M. R., Li, Q., Wan, S., et al. (2020). Security-aware dynamic scheduling for real-time optimization in cloud-based industrial applications. *IEEE Transactions on Industrial Informatics*.

[19] Midya, S., Roy, A., Majumder, K., & Phadikar, S. (2018). Multi-objective optimization technique for resource allocation and task scheduling in vehicular cloud architecture: A hybrid adaptive nature inspired approach. *Journal of Network and Computer Applications, 103*, 58-84.

[20] Moon, Y., Yu, H., Gil, J.-M., & Lim, J. (2017). A slave ants based ant colony optimization algorithm for task scheduling in cloud computing

environments. *Human-centric Computing and Information Sciences*, *7*(1), 1-10.

[21] Pradeep, K., & Jacob, T. P. (2018). A hybrid approach for task scheduling using the cuckoo and harmony search in cloud computing environment. *Wireless Personal Communications*, *101*(4), 2287-2311.

[22] Praveen, S. P., Rao, K. T., & Janakiramaiah, B. (2018). Effective allocation of resources and task scheduling in cloud environment using social group optimization. *Arabian Journal for Science and Engineering*, *43*(8), 4265-4272.

[23] Soltani, N., Soleimani, B., & Barekatain, B. (2017). Heuristic Algorithms for Task Scheduling in Cloud Computing: A Survey. *International Journal of Computer Network & Information Security*, *9*(8).

[24] Sreenu, K., & Malempati, S. (2019). MFGMTS: Epsilon constraint-based modified fractional grey wolf optimizer for multi-objective task scheduling in cloud computing. *IETE Journal of Research*, *65*(2), 201-215.

[25] Sreenu, K., & Sreelatha, M. (2019). W-Scheduler: whale optimization for task scheduling in cloud computing. *Cluster Computing*, *22*(1), 1087-1098.

[26] Srichandan, S., Kumar, T. A., & Bibhudatta, S. (2018). Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm. *Future Computing and Informatics Journal*, *3*(2), 210-230.

[27] Taheri, G., Khonsari, A., Entezari-Maleki, R., & Sousa, L. (2020). A hybrid algorithm for task scheduling on heterogeneous multiprocessor embedded systems. *Applied Soft Computing*, *91*, 106202.

[28] Xavier, V. A., & Annadurai, S. (2019). Chaotic social spider algorithm for load balance aware task scheduling in cloud computing. *Cluster Computing*, *22*(1), 287-297.

[29] Yang, M., Ma, H., Wei, S., Zeng, Y., Chen, Y., & Hu, Y. (2020). A Multi-Objective Task Scheduling Method for Fog Computing in Cyber-Physical-Social Services. *IEEE Access*, *8*, 65085-65095.

[30] Yang, Y., Zhao, S., Zhang, W., Chen, Y., Luo, X., & Wang, J. (2018). DEBTS: Delay energy balanced task scheduling in homogeneous fog networks. *IEEE Internet of Things Journal*, *5*(3), 2094-2106.

[31] Zhang, W., Hu, Y., He, H., Liu, Y., & Chen, A. (2019). Linear and dynamic programming algorithms for real-time task scheduling with task duplication. *The Journal of Supercomputing*, *75*(2), 494-509.

[32] Zheng, W., Wu, H., & Nie, C. (2017). Integrating task scheduling and cache locking for multicore real-time embedded systems. *ACM Sigplan Notices*, *52*(5), 71-80.

Notes on contributors

Anwar R Shaheen, is a Research Scholar from the Department of Computer Science at Alagappa University in India. She has completed Master of Computer Applications and Mphil (Computer Science) in Thassim Beevi Abdul Kader College for Women, Kilakarai. She is currently working as a Vice Principal, Head of the BCA Department in Mohamed Sathak Hamid College, Ramanathapuram. She has 11 years' experience in teaching and 7 years' experience in research. Her major area of research includes Cloud Computing and Computer Networks.

Dr. S. Santhosh kumar received a Master of Information Technology from Bharathidasan University. He received Master of Philosophy in Computer Science from Periyar University and Ph.D. from PRIST University in 2014. He has 16 years of Teaching Experience and 14 Years of Research Experience. His research interests are Computation Biology, Data Analytics, Machine Learning, and Ontology. He has published more than 30 research articles in reputed journals. He is currently working as an Assistant Professor in Computer Science, Alagappa University, Karaikudi, Tamil Nadu India.