International Journal of Information Technology and Web Engineering
Volume 17 • Issue 1

Time Effective Cloud Resource Scheduling Method for Data-Intensive Smart Systems

Jiguang Duan, Institute for Nationalities, Hebei Normal University, China

Yan Li, Shijiazhuang University, China

Liying Duan, Shijiazhuang University, China

Amit Sharma, Jaypee University of Information Technology, India*

ABSTRACT

Cloud computing platforms are being deployed for resource scheduling of real-time data-intensive applications. Cloud computing still deals with the challenge of time-oriented effective scheduling for resource allocation, while striving to provide the efficient quality of service. This article proposes a time prioritization-based ensemble resource management and ant colony-based optimization (ERM-ACO) algorithm in order to aid effective resource allocation and scheduling mechanism which specifically deals with the task group feasibility, assessing and selecting the computing and the storage resources required to perform specific tasks. The research outcomes are obtained in terms of time-effective demand fulfillment rate, average response time, as well as resource utilization time considering various grouping mechanisms based on data arrival intensity consideration. When compared to the present state-of-the-art methods, optimal fitness percentage of 98% is observed, signifying the feasible outcomes for real-time scenarios.

KEYWORDS

Big Data Processing, Cloud Computing, Data-Intensive Applications, Real-Time Experimentation, Resource Scheduling, Time-Effective

INTRODUCTION

Cloud computing has evolved as an important aspect of network sharing computational model, involving the applicability of various commercial resources in the current scenario. This technological advent has revolutionized the cloud computing as the third technology in the last few decades (Guo et al., 2008; Fenz et al. 2020). With the emergence of data-intensive applications, cloud computing has become the most widely used virtualization platform for the complex applications (Greenberg et al., 2008, Ijaz et al., 2013, Wang and Ng, 2010; Rajan, 2020). The various types of applications including cloud computing are web applications, distributed applications like e-commerce, etc. This technology has emerged as a new paradigm for providing the reliable service in the current dynamic environment. The basic concept of cloud computing relies on the storage strategy that is designed to store the user data on internet instead of storing it locally (Kumar et al., 2017). The rapid improvement

DOI: 10.4018/IJITWE.306915 *Corresponding Author

in energy consumption of the internet-based data, time effectiveness has become a major factor for improving its performance (Srichandan et al., 2018). The time effectiveness of a cloud computing-based application is affected by communication capacity, multi-tasking execution and various other factors (Natesan and Chokkalingam, 2019). Thus, the overall performance of cloud computing aspects can be improved using task scheduling strategies.

The task scheduling and resource allocation strategies should be optimized for precise and accurate resource provisioning and flexible execution of cloud computing. There are various optimized resource allocation and task scheduling mechanisms reported in the literature, contributing to the cloud computing-based software engineering frameworks (Chen et al., 2015). However, there are various possibilities for optimizing new strategies as the current scheduling algorithms are costly and are prone to errors as well. The various studies suggested that in order to accomplish the cloud computing-based task execution, there are various aspects which should be taken care of:

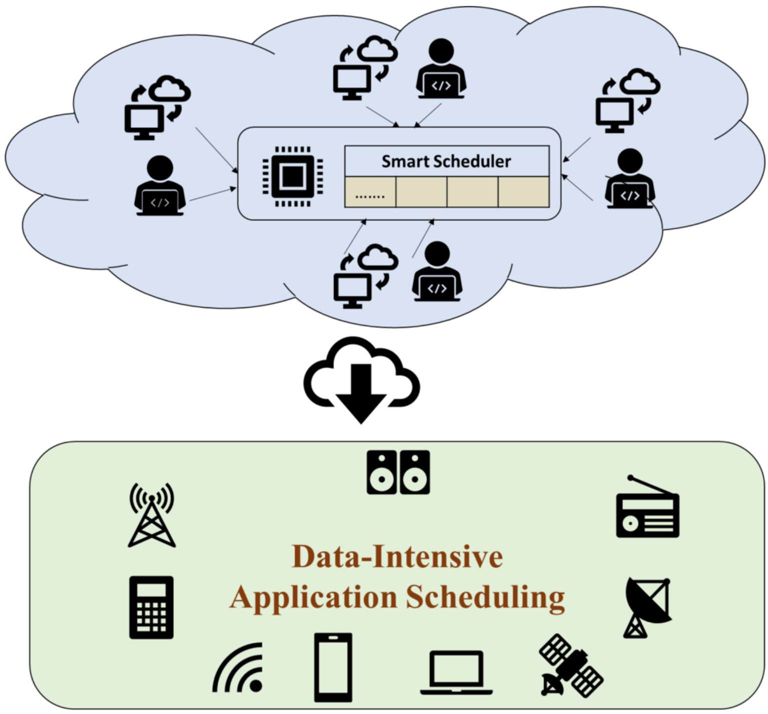
- Resource distribution and task allocation is important.
- Order of task execution by the clouds in order to avoid data dependencies.
- Overhead scheduling for switching the task sequence and planning.
- Cost reduction to meet the increase demand of computer resources.
- Efficient allocation as well as deallocation of resources for proper resource utilization.
- Flexible scaling speed for effectual infrastructure utilization.
- Easy and flexible accessibility at any convenient time around the world.

These aspects are needed to be effectually considered for effective resource allocation and task scheduling which is very important for high performance computing and also plays a vital role in embedded systems (Min-Allah et al., 2020). The basic task scheduling methodology for cloud computing process in depicted in Figure 1 (Hu et al., 2020).

The rapid development in the generation of data has made the expansion of big data analytics that commonly utilizes the higher volume data with diverse variety and velocity. The various challenges of big data capabilities like its storage, management and efficiency can be eradicated using the concept of cloud computing (Tomar et al., 2018; Singhal et al., 2016). The cloud paradigm is able to manage the geographically distributed big data which is beneficial for its efficient processing and real-time analysis. the big data analysis is specifically empowered by machine learning, statistical algorithms as well as optimization approaches for efficient processing and low latency. Although, the processing of big data using cloud computing concept provides higher efficiency as they directly stream the data to cloud platform for real time processing. But the drawback lies in data dependency issue as the processing of big data involves various stages and the task processing at the subsequent stage require the output data from the previous stage. The workflow of big data processing is both highly computational as well as data intensive. However, the task scheduling for heterogeneous environment requires better algorithms for the allocation of resources as well as task scheduling. There have been various investigations involving resource heterogeneity within the clouds given by Qureshi et al., (2020) but still there is a large scope of improvement in analysis of multi-objective intelligent optimization algorithms for time effective scheduling and resource allocation.

This article proposes an effective resource allocation and scheduling mechanism for intelligent management of cloud computing workflow during the execution of data-intensive applications. Time prioritization is the main focus of this research for which, this article exploits a dynamic resource prediction strategy which uses the ensemble resource management and Ant Colony based optimization (ERM-ACO). The novelty of the significant contribution of this article lies in the utilization of dynamically optimized cloud scheduling mechanism which can achieve effective resource scheduling. This novel approach deals with various aspects of resource scheduling by engaging efficient distribution, allocation and deallocation of resources for proper resource utilization. The data dependencies are avoided in the proposed approach by proper ordering of task execution while

Figure 1. Task scheduling methodology using cloud computing



using the grouping mechanism and task sequencing. Planning and cost reduction is done in order to meet the increasing demand of computer resources. Flexible accessibility along with improvement in scaling speed for effectual infrastructure utilization is done by utilizing the open-source J-Storm platform for real-time experimentation utilizing the big data processing in case of dynamic resource scheduling. J-Storm is a productive big data-based resource management application that includes the details of voluminous activities of several tasks including the task progress updates, resource deployment, etc. which other platforms cannot assess for real-time dynamic resource scheduling. Thus, this platform is utilized in the proposed work in order to enable the resource management and execution. The proposed (ERM-ACO) framework is also compared with the present state-of-the-art methods like particle swarm optimization (PSO), genetic algorithm (GA) and evolutionary algorithm (EA) and feasibility of the algorithm is validated utilizing various strategical parameters like time-effective demand fulfillment rate, average response time and resource utilization time. The reliable time-effective scheduling is achieved using the proposed approach and the comparative analysis revealed its efficacy to perform better in the real-time scenarios.

The rest of this article is structured as: Literature review is provided in section 2 which presents the current state-of-the-art work in the field of resource allocation and scheduling. The material and methods utilized for this research work are provided in section 3 followed by the results and discussion of the experimentation in section 4. Section 5 of this article provides the concluding remarks of this research work along with the future research indications.

LITERATURE REVIEW

The data intensive smart systems have become an active research area due to the utilization of big data in the current technological advent. Various researchers investigated the use of big data technology in the field of healthcare, task scheduling, resource allocation, etc. (Luo et al., 2016). In the recent years, various researchers are using the big data-based methodologies for inventing new techniques and discoveries for resource allocation. However, resource allocation is a difficult task due to increasing data complexity as well as due to the impact of energy consumption. It is required to maintain a trade-off among the resource allocation and performance. Various heuristic and metaheuristic approaches have been adopted by the researchers to provide better solutions for task allocation.

Lee and Zomaya (2012) utilized a heuristic approach for maximum resource allocation and also provide efficient outcomes in terms of cost as well as least energy consumption. Nosrati and Karimi (2016) accomplished the resource allocation task by identifying the latencies in communication and geographical distance of the system. This approach utilizes the optimization methodology for minimizing the energy consumption losses. Dynamic resource allocation method was presented by Verma et al. (2016), also exploited the dynamic resource prediction and allocation framework that avoids the computational time and cost while improving the resource performance and utilization. Yang et al. (2020) proposed a modified resource allocation scheme that manages the energy consumption. Xiao et al. (2012) on the other hand provided a new mechanism for resource allocation while overcoming the system workload and minimizing the energy consumption. Netjinda et al. (2012) presented a provisioning method for resource allocation that uses the Particle Swarm Optimization (PSO) method for cost minimization which characterizes the resource configuration and decision. Ding et al. (2014) evaluated the performance of resource allocation based on the resource quality of service recommendations. Aminikhanghahi and Cook (2017) also presented an approach for resource provisioning that reduces the cost of execution along with time effectiveness measure. They utilized the k-mean clustering approach for the identification and categorization of resources. A feasible multi-objective framework for resource allocation was established by Liu et al. (2018) for the performance evaluation, resource utilization as well as effectual energy consumption. A multi-agent system was characterized by Da La Prieta, et al. (2019) which enables the effective allocation of resources using the cloud computing environment with global coordination. Tseng et al. (2020) investigated the scheduling issues and studied the role of trusted nodes in resource scheduling. It was revealed by the authors that resilience is maintained by the introduction of trusted nodes, while reducing the latency and energy consumption. The researchers Ali et al. (2020) analysed the drawbacks of traditional centralized systems for the allocation of resources, resource optimization as well as its maintenance. It was revealed by that the scalability and robustness of the system can be improved while effectively reducing the occurrence of failure, by utilizing decentralized management.

An effectual strategy for resource allocation and scheduling using the cloud platform was proposed by Maheshwari et al. (2021) addressing the NP-complete problem using a number of meta heuristic algorithms and a reinforces learning mechanism. This methodology maximises the cloud throughput and eventually minimizes the completion time and production cost. The review and future research direction for the reinforced learning methods in the field of resource scheduling was presented by Zhou et al. (2021). They focused their research work in the direction of deep learning and reinforcement learning for considering the resource scheduling prospect of cloud computing. They further reviewed the applications of reinforcement highlighting the major challenges in this scenario. The multi-dependency scenario of resource scheduling is assessed by Prakash et al. (2021). They proposed a parent to child node dependency-based scheduling algorithm emphasizing proper resource scheduling overcoming the limitations posed by time-based schedulers. This algorithm is applicable to the scientific applications like cybershake, montage, etc. and the simulation based workflow scheduling provides satisfactory performance. Malik et al. (2021) studied the varying parameters and objectives for addressing the energy consumption problem. Effectual resource utilization is accomplished utilizing task classification and thresholding which is further followed by PSO optimization technique in order to obtain the best scheduler. The effectiveness of the approach is justified in terms of various parameters like energy consumption, load balancing, etc. The service level agreement was satisfied by Singh et al. (2021) in order to upgrade the performance of task scheduling in cloud computing scenario.

The QOS based resource allocation is done using the swarm-based ACO in order to simulate the cloud environment. Better outcomes are achieved in terms of various QOS parameters and real time problem is solved while observing effectual reduction in the cost execution and resource utilization parameters. Yu, (2021) aimed at solving the resource scheduling and optimization problem by utilizing an ensemble fuzzy clustering cloud resource scheduling algorithm. Two types of scheduling is done by the authors, one is the task scheduling and the other is the user resource scheduling. The malicious nodes are eradicated for this scheduling mechanism and the trust sensitive nodes are only selected in order to provide effective cloud resource allocation. The throughput of the entire cloud system is improved by embodying the advantages of cloud resources for the satisfaction of cloud-based users.

From the literature survey, it has been observed that the most of the existing energy efficiency models consider two main parameters: time effective resource scheduling and minimized energy consumption. The researchers are also seeking for providing the best suitable framework which can satisfy all the constraints of data-intensive applications like engagement of efficient resource distribution, allocation and deallocation, avoidance of data dependencies, task sequencing, planning and cost reduction. However, it was revealed from the literature survey that for these data-intensive applications, the traditional models do not provide suitable response in the present scenarios. Thus, this situation led the foundation of developing an optimization-based approach for effective resource allocation and scheduling management.

PROPOSED METHODOLOGY

This section presents the details of the material and methods utilized for time-effective resource scheduling, the details of simulation environment and specifications of performance evaluation indices.

Ant Colony Optimization

The conventional Ant Colony optimization methodology is basically inspired from the real ant colonies which specifically searches for the shortest path between the colony and the food. They shed the pheromones on their path to the food source and the intensity of that determines the recognition of the shortest path which further increases the passage of ants through that particular path. For solving the challenge of resource management and task scheduling, this ACO algorithm can be utilized considering the scheduling of independent tasks in a cloud arrangement and the number of ants considered should be less than or equal to the number of tasks to be allocated. However, the conventional state-of-the-art metaheuristic ACO algorithm is not used in this application as the proposed ERM-ACO algorithm deals with various aspects efficient resource distribution, allocation and deallocation. The avoidance of data dependencies is addressed in the proposed approach by effectual ordering of task execution while utilizing the grouping mechanism while using the aspect of task sequencing, planning and cost reduction to meet the increasing demand of computer resources.

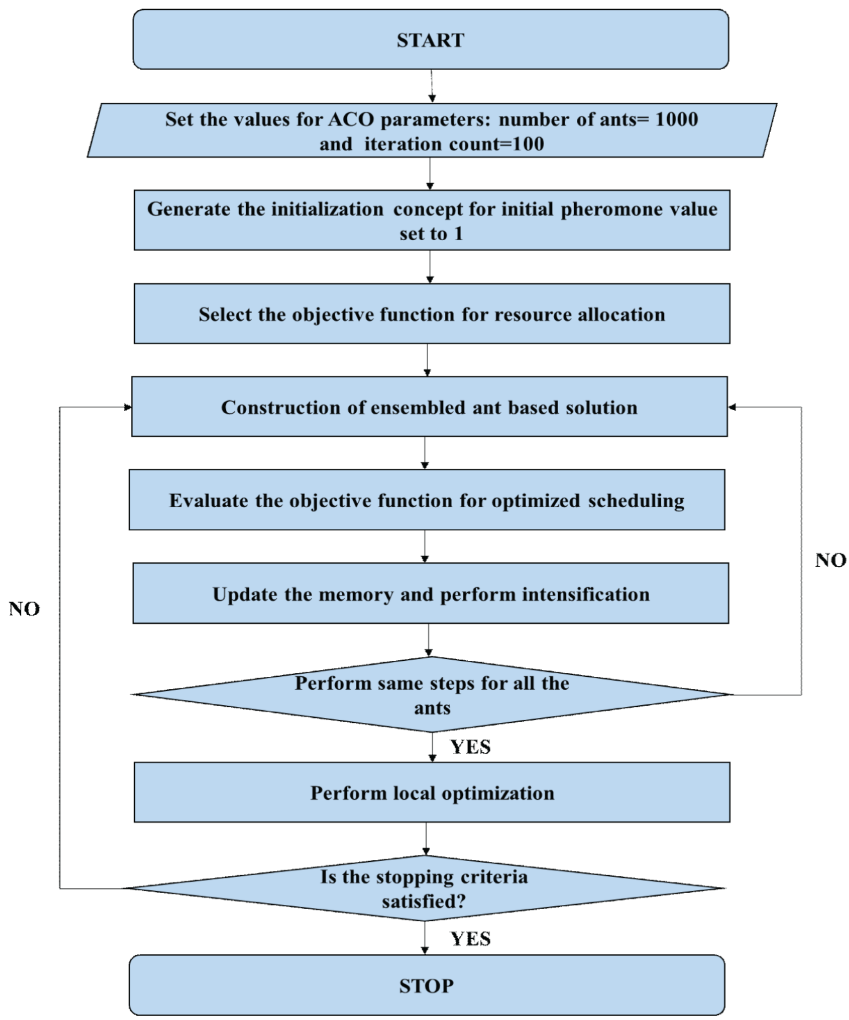
Ensembled Resource Management and Ant Colony Based (ERM-ACO) Algorithm

The proposed ensemble resource management and ant colony optimization algorithm (ERM-ACO) for resource allocation and scheduling while keeping the time-effective constraint in mind. The ant ERM-ACO algorithm flowchart is depicted in Figure 2 describing the entire idea of the optimization process.

The major objective of the proposed ERM-ACO algorithm is to perform resource allocation based heuristic optimization in order to provide a reliable web service employing cloud computing approach.

In the proposed methodology, each of the individual ant starts with a random task (T_x) and the resource (R_y) which is needed to be allocated for the processing of this arbitrary task. Then the probable function is assigned in order to execute the task T_x using the computed number of resources. Thus, the step-by-step execution of the task is done and the ants are built up in order to obtain the solution of task assignment using limited number of resources. The initial value set for the pheromone

Figure 2. Flowchart of ERM-ACO Algorithm



should be a positive constant integer ‘1’ and this value further modified by the ants at the end of each iteration. The best solution is thus formulated from the minimum and maximum value of the objective function. The optimal solution is obtained after completing all the iterations. The complete algorithmic process is provided in the following algorithm.

Time prioritization is the major concern of this work, therefore the algorithm exploited utilizes a dynamic resource prediction strategy which uses the ensembled resource management and Ant Colony based optimization. This algorithm governs the task schedulable scenario for real-time tasks with difference in data optimality having varying timing constrictions. The execution procedure for this algorithm comprises of initially checking the task group feasibility, then assessing and selecting the computing and the storage resources required to perform the task. All these criteria should be met under the minimum execution time and should meet the defined deadlines. The schedule of all the groups should be such that it should take minimum time and should be competed with cost effectiveness as well. If these conditions are satisfied by all the optimized schedules, then the stopping

Table 1. Algorithm 1: ERM-ACO algorithm

Algorithm: ERM-ACO Algorithm
Input: Task workflow to be managed
Output: Managed optimal solution for the task workflow execution

Initialize the ACO parameters: number of ants= 1000, iteration count=100 and initial pheromone count to 1.
While $R_y < T_x$, iteration=1
    for each individual ant
        for each individual task
        Compute the objective function using the equation: $P_{x,y} = \frac{[R_{x,y}][T_{x,y}]}{\sum_{x=1}^{n}[R_{x,y}][T_{x,y}]}$, where $P_{x,y}$ is the transition probability from x to y and R indicates the resource and T are the tasks.
        if the tasks are allocated in the same layer
        end for
        Update the local pheromone and perform intensification using the equation: $\Delta\tau_{xy} = D / clock_{xy}$ where D is the constant, $clock_{xy}$ indicates the local task completion time.
        else
        Adjust the local pheromone
    end for
  Obtain the optimal solution for this particular iteration
  iteration++
 Update the global pheromone value using the equation: $\Delta\tau_{xy}, = D / bestclock_{xy}$ where D is the constant, $bestclock_{xy}$ indicates the global task completion time.
End while
If the stopping criteria is satisfied, obtain the optimal solution for task allocation in the workflow.

criteria are satisfied and the resource allocation procedure is completed. This time prioritization is experimentally analyzed using various grouping mechanisms for big data processing and dynamic resource scheduling to validate the reliability of the system.

Simulation Environment

The simulation environment in which the proposed ERM-ACO algorithm is implemented comprises of six different physical nodes which are configured at different configurations. These six physical nodes consist of one of the master node, four computer nodes and one remaining node that act as a client for simulating the varying user requests. However, this is not the default specific criteria and the number of computer nodes can be increased or decreased depending upon the task workflow. As the number of tasks increases, number of notes can be increased, however, the master node and the client node should be constant. But, upon increasing the number of computer nodes, system complexity increases. Therefore, for this particular application of resource scheduling and task execution, six node system is considered to improve the time efficacy while not compromising the system complexity.

The complete simulation environment configuration along with the detail of resource type and its configuration is provided in Table 2. This article uses the open-source platform called J-Storm for real-time experimentation for dynamic resource scheduling due to its flexible accessibility along with scaling speed for effectual infrastructure utilization for real-time experimentation utilizing the real-time dynamic resource scheduling and big data processing.

Table 2. Simulation environment detail

Type of Resource	Configuration Details
Hardware	Intel (R) CPU with i5 processor
Software	8GB RAM, 1TB external storage, ethernet port, Maven 3.2.2 Compiler environment.

Performance Evaluation Indices

The performance evaluation of the proposed algorithm is done using various indices like time-effective demand fulfillment rate, response time and resource utilization time:

1. **Time-effective demand fulfillment rate:** This evaluation index is measured as the ratio of resources allocated successfully within the deadline to the total number of resource allocation requests made during that particular period of time. The time-effective demand fulfillment rate is expressed by Eq. (1):

$$T_{QOS} = \frac{RA_{success}\left(T\right)}{RQ_{Total}\left(T\right)} \tag{1}$$

where $RQ_{Total}\left(T\right)$ represents the total resource allocation requests made during time period (T) and $RA_{success}\left(T\right)$ represents the resources allocated successfully within the time limit (T).

2. **Response time:** It is expressed as the difference between the resource allocation time of the application to the time at which the resource allocation is completely process. Response time is defined as the average time taken for successful processing of the resource allocation task. The expression for response time is given by Eq. (2):

$$T_{Response} = \frac{\sum_{i=0}^{n\left(T_R\right)} D_i}{n\left(T_R\right)} \tag{2}$$

where T_R expresses the current rth time slot, $n\left(T_R\right)$ expresses the total amount of resource allocation in time period T_R, i represents the initial data point and D_i expresses the time delay during the processing of ith data point during the time period T_R.

3. **Resource utilization time:** This evaluation index is expressed by the averaged ratio of actual resource usage to the exact resources allocated in the particular time frame. The calculation formula for resource utilization is expressed as Eq. (3):

$$R_{Utilization}\left(T\right) = \frac{1}{n}\sum_{f=1}^{n}\frac{RA_{usage}\left(T_f\right)}{RA_{Allocated}\left(T_f\right)} \tag{3}$$

where $RA_{usage}\left(T_f\right)$ represents the resource usage, $RA_{Allocated}\left(T_f\right)$ indicates the resource allocation during the particular time frame f average over the total number of resources n.

RESULTS AND DISCUSSION

The experimental investigation is done in three-tier scenario. Initially, the performance of the proposed methodology is tested for all the performance evaluation indices based in the data grouping mechanism. Then the time-effectiveness for varying resource configuration is evaluated followed by the task scheduling analysis and comparison with state-of-the-art methods.

Performance Analysis Based on Data Grouping Mechanism

In this experimentation, the data from the J-Storm platform is partitioned into five different groups based on the average arrival rate of the data packages. The J-storm productive big data data-based resource management platform includes the details of voluminous activities of several tasks including the task progress updates, resource deployment, etc. for assessing the real-time dynamic resource scheduling practices. This is the major reason for using the data from this particular platform for experimental analysis. The data grouping is done using the lowest data arrival intensity of 450 tps/s to the highest intensity of 2150 tps/s. However, more or less groups can be formed in this range but the authors have considered five groups considering an average change in the data arrival intensities. This data grouping mechanism is useful in identifying the accurate time-effective demand fulfillment rate based on the arrival strength. The tabular representation of grouping mechanism is given in Table 2.

Table 3 represents the grouping value corresponding to the average number of tuples per second (tps/s). The default resource configuration value for the J-Storm cluster is considered for initial resource allocation within the time guarantee requirement of 500 milliseconds. The time-effective demand fulfillment rate is thus computed in order to indicate the satisfaction of time-sensitive requirements. Figure 3 indicates the analysis of time-effective demand fulfillment rate under varying data arrival strength.

The graphical representation reveals that the time-effective demand fulfillment rate under varying data arrival strength is ensured using the proposed methodology for big data processing. The time-effective demand fulfillment is maintained at the initial grouping as well as same improvement is noticed when the application load attained the peak value. The proposed approach dynamically allocates the sufficient number of resources for the cloud platform so that the sufficient amount of storage capacity is available at the peak of the resource demand as well. Therefore, it is revealed that as the load demand for the resources is reduces, the demand fulfillment is achieved in the appropriate time. Another important factor ensuring the timeliness of big data processing is response time. This parameter is also important for indicating the time-effectiveness of the proposed scheme. Average response time for different grouping mechanisms is compared in Figure 4.

Table 3. Grouping mechanism based on data arrival strength

Grouping Mechanism	Data arrival intensity consideration
Group 1	450 tps/s
Group 2	860 tps/s
Group 3	1150 tps/s
Group 4	2010 tps/s
Group 5	2150 tps/s

Figure 3. Time-effective demand fulfilment rate under varying data arrival strength

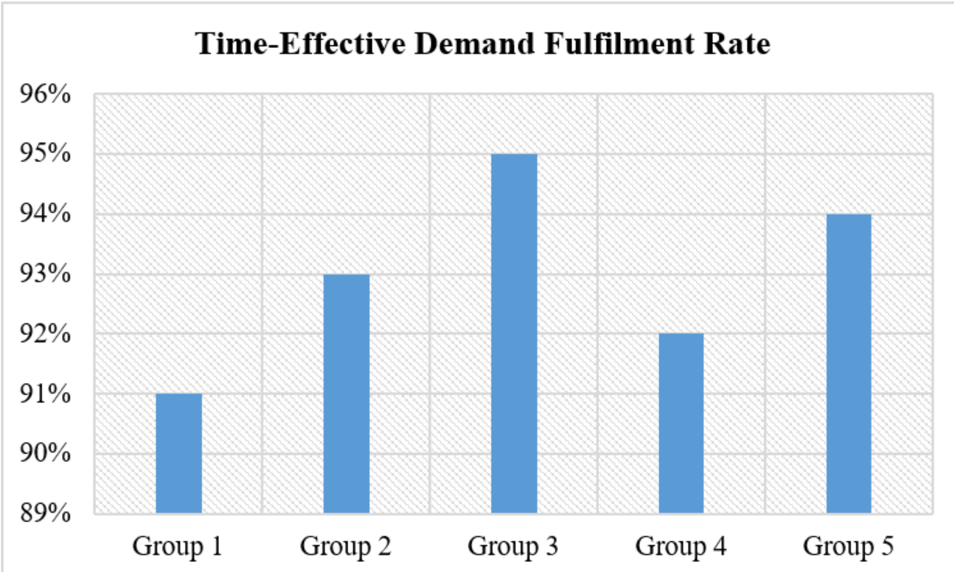


Figure 4. Average response time for different grouping mechanisms

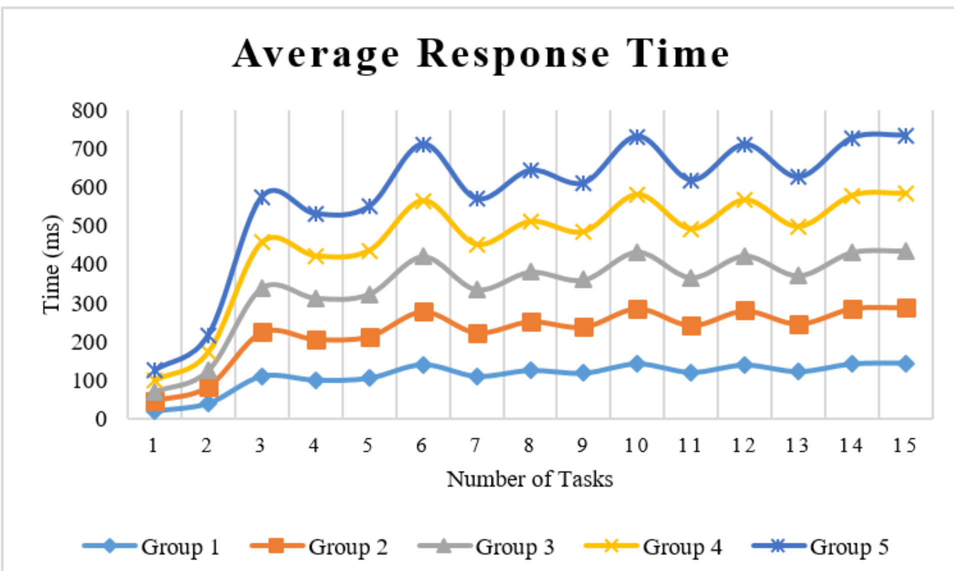
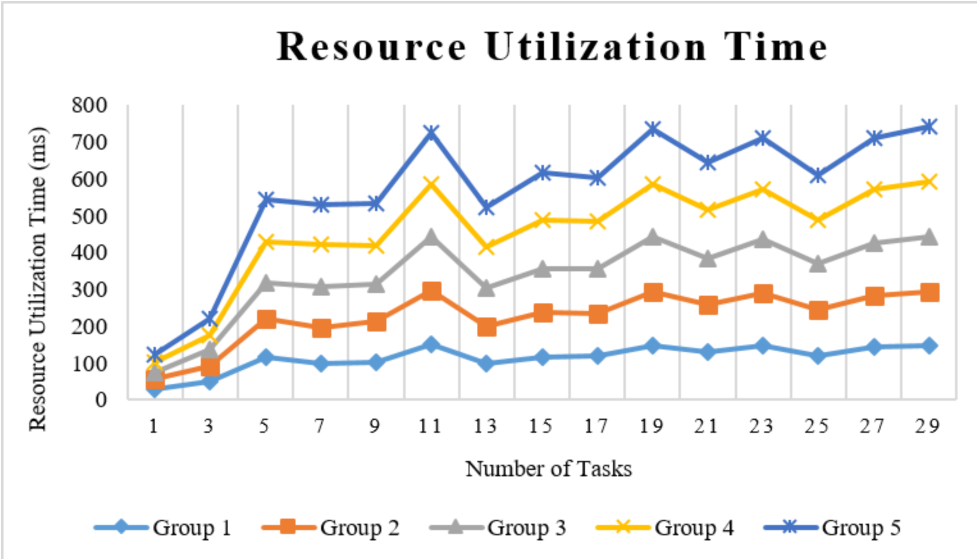


Figure 4 depicts the improvement in response time for different grouping mechanisms. The interaction among the computing as well as storage resources is avoided in this system using the multiple computing resource containers. Further a comparison in drawn in terms of resource utilization time for different groups and its observation is indicated in Figure 5.

This sort of dynamic allocation led to the reduction in the amount of data queued for each of the component, leading to an increase in resource allocation during the peak load time and minimizing the

Figure 5. Resource utilization time for different grouping mechanisms



dwell time. All of these parametric evaluations also lead to the reduction in the complete processing time which in turn reduces the overall time elapsed between the data flow into the cloud computing system to the complete processing time.

Task Scheduling Analysis and Comparison With State-of-the-Art Methods

The outcomes of ERM-ACO based task scheduling analysis are compared with the other two state-of-the-art methods in terms of overall resource utilization and optimal fitness during varying number of quality services. One method considered for state-of-the-art comparison is particle swarm optimization (PSO) approach, genetic algorithm (GA) and evolutionary algorithm (EA). The population size considered for comparison is 1000 and the service class varies from 1 to 50 per user. Figure 6 provides the comparison of overall resource utilization percentage for different state-of-the-art methods.

This comparative study indicates the overall resource utilization for different number of tasks. It is revealed that the performance of the proposed ERM-ACO algorithm remains somewhat stable, whereas, the performance of PSO, GA and EA algorithms initially indicates upward trending with the sharp fall towards the end. The performance of the proposed ERM-ACO algorithm increases gradually with the number of tasks involved. The comparative analysis in terms of optimal fitness during varying number of quality services is provided in Figure 7.

The comparison provided in Figure 7 reveals that the association rate between the optimal fitness of ERM-ACO algorithm and the actual optimal fitness curve is observed to be around 98%. However, for other algorithms like PSO, GA and EA this rate is observes as 89% and 85% respectively. The dynamic combination of fitness curve is depicted by the gradual increase in optimal fitness level with the increase in the number of quality services provided. The observations made from this comparison reveals that the ERM-ACO algorithm performs better when the number of tasks are increased, thus, showing its competence for solving the large-scale optimization problems. Therefore, all the constraints of resource utilization are satisfied using the ERM-ACO algorithm, however, PSO, GA and EA algorithms does not guarantee the feasible and generalized solutions.

Figure 6. Comparative Analysis of overall resource utilization percentage for state-of-the-art methods

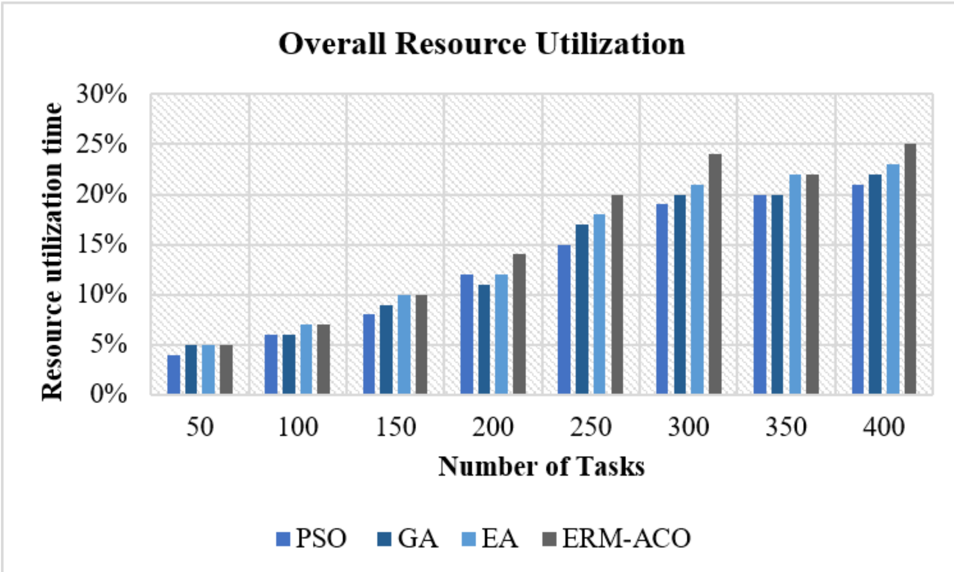
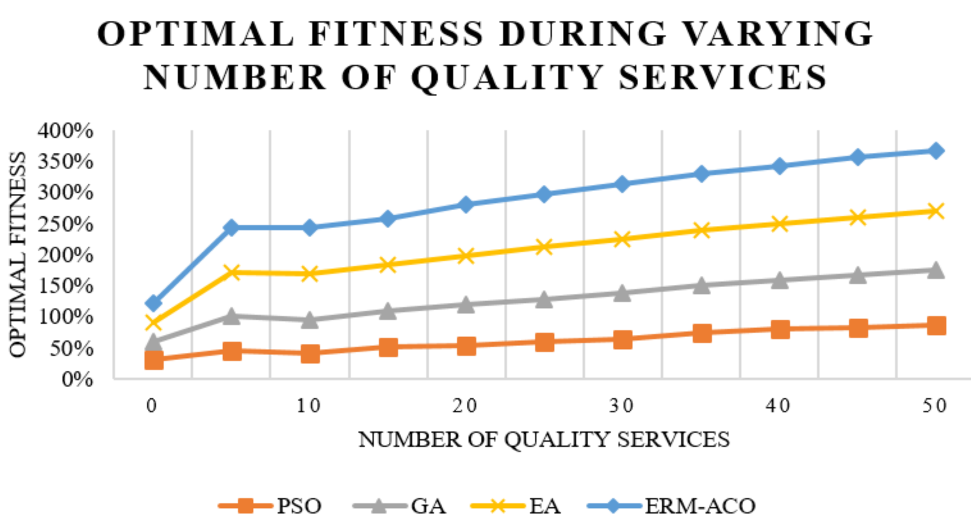


Figure 7. Comparative Analysis of optimal fitness during varying number of quality services for state-of-the-art methods



CONCLUSION

This article proposes an ensembled resource management and Ant Colony based optimization (ERM-ACO) algorithm for time effective resource allocation and scheduling. The J-Storm open-source platform is utilized in this article for flexible accessibility and effectual infrastructure utilization during real-time experimentation using the big data processing. The research outcomes are observed in terms of time-effective demand fulfillment rate, average response time as well as resource utilization time. For the experimentation a grouping mechanism is considered which is based on based on data arrival intensity considerations. The proposed framework when compared to the present state-of-the-art methods in terms of resource utilization and optimal fitness value. The overall resource utilization percentage for the proposed framework is observed maximum 25% which is way better comparative to the other state of the art methods like PSO and GA algorithms. The optimal fitness percentage of 98%is observed for ERM-ACO methodology which is far better than 89% and 85% optimization fitness values obtained for state-of-the-art PSO, GA and EA respectively. ERM-ACO algorithm performs better and it establishes its competence for solving the large-scale optimization problems providing a feasible and generalized solution for real time data intensive applications. This work can be extended to other grouping mechanisms in the future part of this research, to reveal the effectiveness of the approach for varying cloud platforms and different computer configurations.

FUNDING AGENCY

The publisher has waived the Open Access Processing fee for this article.

REFERENCES

Ali, F. S., Aloqaily, M., Alfandi, O., & Ozkasap, O. (2020). Cyberphysical blockchain-enabled peer-to-peer energy trading. *Computer*, *53*(9), 56–65. doi:10.1109/MC.2020.2991453

Ali, F., Bouachir, O., Ozkasap, O., & Aloqaily, M. (2020). Synergy Chain: Blockchain-assisted adaptive cyberphysical P2P energy trading. *IEEE Transactions on Industrial Informatics*, *17*(8), 5769–5778. doi:10.1109/TII.2020.3046744

Aminikhanghahi, S., & Cook, D. J. (2017). A survey of methods for time series change point detection. *Knowledge and Information Systems*, *51*(2), 339–367. doi:10.1007/s10115-016-0987-z PMID:28603327

Chen, Z. G., Du, K. J., Zhan, Z. H., & Zhang, J. (2015). Deadline constrained cloud computing resources scheduling for cost optimization based on dynamic objective genetic algorithm. *2015 IEEE Congress on Evolutionary Computation (CEC)*, 708-714. doi:10.1109/CEC.2015.7256960

De la Prieta, F., Rodríguez-González, S., Chamoso, P., Corchado, J. M., & Bajo, J. (2019). Survey of agent-based cloud computing applications. *Future Generation Computer Systems*, *100*, 223–236. doi:10.1016/j.future.2019.04.037

Ding, S., Xia, C., Cai, Q., Zhou, K., & Yang, S. (2014). QoS-aware resource matching and recommendation for cloud computing systems. *Applied Mathematics and Computation*, *247*, 941–950. doi:10.1016/j.amc.2014.09.058

Fenz, T., Foerster, K. T., Schmid, S., & Villedieu, A. (2020). Efficient non-segregated routing for reconfigurable demand-aware networks. *Computer Communications*, *164*, 138–147. doi:10.1016/j.comcom.2020.10.003

Greenberg, A., Hamilton, J., Maltz, D. A., & Patel, P. (2008). The cost of a cloud: Research problems in data center networks. *Computer Communication Review*, *39*(1), 68–73. doi:10.1145/1496091.1496103

Guo, C., Wu, H., Tan, K., Shi, L., Zhang, Y., & Lu, S. (2008). Dcell: A scalable and fault-tolerant network structure for data centers. *Computer Communication Review*, *38*(4), 75–86. doi:10.1145/1402946.1402968

Hu, Y., Wang, H., & Ma, W. (2020). Intelligent cloud workflow management and scheduling method for big data applications. *Journal of Cloud Computing*, *9*(1), 1–13.

Ijaz, S., Munir, E. U., Anwar, W., & Nasir, W. (2013). Efficient scheduling strategy for task graphs in heterogeneous computing environment. *The International Arab Journal of Information Technology*, *10*(5), 486–492.

Kumar, D., Kumar, S., Bansal, R., & Singla, P. (2017). A survey to nature inspired soft computing. *International Journal of Information System Modeling and Design*, *8*(2), 112–133. doi:10.4018/IJISMD.2017040107

Lee, Y. C., & Zomaya, A. Y. (2012). Energy efficient utilization of resources in cloud computing systems. *The Journal of Supercomputing*, *60*(2), 268–280. doi:10.1007/s11227-010-0421-3

Liu, D., Sui, X., Li, L., Jiang, Z., Wang, H., Zhang, Z., & Zeng, Y. (2018). A cloud service adaptive framework based on reliable resource allocation. *Future Generation Computer Systems*, *89*, 455–463. doi:10.1016/j.future.2018.05.059

Luo, J., Wu, M., Gopukumar, D., & Zhao, Y. (2016). Big data application in biomedical research and health care: A literature review. *Biomedical Informatics Insights*, *8*, 1–10. doi:10.4137/BII.S31559 PMID:26843812

Maheshwari, S., Shiwani, S., & Choudhary, S. S. (2021, March). The Efficient Resource Scheduling Strategy in Cloud: A Metaheuristic Approach. *IOP Conference Series. Materials Science and Engineering*, *1099*(1), 012027. doi:10.1088/1757-899X/1099/1/012027

Malik, N., Sardaraz, M., Tahir, M., Shah, B., Ali, G., & Moreira, F. (2021). Energy-Efficient Load Balancing Algorithm for Workflow Scheduling in Cloud Data Centers Using Queuing and Thresholds. *Applied Sciences (Basel, Switzerland)*, *11*(13), 5849. doi:10.3390/app11135849

Min-Allah, N., Qureshi, M. B., Jan, F., Alrashed, S., & Taheri, J. (2020). Deployment of real-time systems in the cloud environment. *The Journal of Supercomputing*, *77*(2), 2069–2090. doi:10.1007/s11227-020-03334-7

Natesan, G., & Chokkalingam, A. (2019). Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm. *ICT Express*, *5*(2), 110–114. doi:10.1016/j.icte.2018.07.002

Netjinda, N., Sirinaovakul, B., & Achalakul, T. (2012). Cost optimization in cloud provisioning using particle swarm optimization. In *9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology* (pp. 1-4). IEEE.

Nosrati, M., & Karimi, R. (2016). Energy efficient and latency optimized media resource allocation. *International Journal of Web Information Systems*, *12*(1), 2–17. doi:10.1108/IJWIS-10-2015-0031

Prakash, V., Bawa, S., & Garg, L. (2021). Multi-Dependency and Time Based Resource Scheduling Algorithm for Scientific Applications in Cloud Computing. *Electronics (Basel)*, *10*(11), 1320. doi:10.3390/electronics10111320

Qureshi, M. S., Qureshi, M. B., Fayaz, M., Zakarya, M., Aslam, S., & Shah, A. (2020). Time and Cost Efficient Cloud Resource Aallocation for Real-Time Data-Intensive Smart Systems. *Energies*, *13*(21), 5706. doi:10.3390/en13215706

Rajan, C. D. S. (2020). Design and implementation of fuzzy priority deadline job scheduling algorithm in heterogeneous grid computing. *Journal of Ambient Intelligence and Humanized Computing*, 1–8.

Singh, H., Bhasin, A., & Kaveri, P. R. (2021). QRAS: Efficient resource allocation for task scheduling in cloud computing. *SN Applied Sciences*, *3*(4), 1–7. doi:10.1007/s42452-021-04489-5

Singhal, A., & Tomar, R. (2016, October). Intelligent accident management system using IoT and cloud computing. In *2016 2nd international conference on next generation computing technologies (NGCT)* (pp. 89-92). IEEE. doi:10.1109/NGCT.2016.7877395

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. doi:10.1016/j.fcij.2018.03.004

Tomar, R., Khanna, A., Bansal, A., & Fore, V. (2018). An architectural view towards autonomic cloud computing. In *Data Engineering and Intelligent Computing* (pp. 573–582). Springer. doi:10.1007/978-981-10-3223-3_55

Tseng, L., Wu, Y., Pan, H., Aloqaily, M., & Boukerche, A. (2020). Reliable broadcast with trusted nodes: Energy reduction, resilience, and speed. *Computer Networks*, *182*(4), 107486. doi:10.1016/j.comnet.2020.107486

Verma, M., Gangadharan, G. R., Narendra, N. C., Vadlamani, R., Inamdar, V., Ramachandran, L., Calheiros, R. N., & Buyya, R. (2016). Dynamic resource demand prediction and allocation in multitenant service clouds. *Concurrency and Computation*, *28*(17), 4429–4442. doi:10.1002/cpe.3767

Wang, G., & Ng, T. E. (2010). The impact of virtualization on network performance of amazon EC2 data center. In 2010 Proceedings IEEE INFOCOM (pp. 1-9). IEEE.

Xiao, Z., Song, W., & Chen, Q. (2012). Dynamic resource allocation using virtual machines for cloud computing environment. *IEEE Transactions on Parallel and Distributed Systems*, *24*(6), 1107–1117. doi:10.1109/TPDS.2012.283

Yang, J., Jiang, B., Lv, Z., & Choo, K. K. R. (2020). A task scheduling algorithm considering game theory designed for energy management in cloud computing. *Future Generation Computer Systems*, *105*, 985–992. doi:10.1016/j.future.2017.03.024

Yu, J. (2021). Qualitative Simulation Algorithm for Resource Scheduling in Enterprise Management Cloud Mode. *Complexity*, *2021*, 2021. doi:10.1155/2021/6676908

Zhou, G., Tian, W., & Buyya, R. (2021). *Deep Reinforcement Learning-based Methods for Resource Scheduling in Cloud Computing: A Review and Future Directions.* arXiv preprint arXiv:2105.04086.

Jiguang Duan is associated with Institute for Nationalities Attached to Hebei Normal University. Research interests are data analytics. Works at Shijiazhuang University Shijiazhuang, Hebei, China.

Liying Duan is associated with Shijiazhuang University Shijiazhuang, Hebei. Research area is Cloud computing.