Received: July 24, 2018

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

An Improved Resource Allocation Model for Grid Computing Environment

Anju Shukla[1]* **Shishir Kumar[1]** **Harikesh Singh[1]**

[1]*Department of Computer Science and Engineering*
Jaypee University of Engineering and Technology, Guna, MP, India
* Corresponding author's Email: anjushukla.iiitb@gmail.com

Abstract: Grid is collection of heterogeneous resources which are highly dynamic and unpredictable in nature. Various resource allocation techniques are used for effective use of resources. Tasks are submitted with constraints (cost, deadline, etc.) to improve the performance of distributed environment. Based on resource availability, several tasks select the same resource to achieve minimum execution cost. Therefore, an effective scheduling is required to execute tasks on the same resource to reduce average waiting time of tasks. In this paper, an extensive survey of resource allocation techniques is done based on several parameters. A resource allocation and task scheduling mechanism is proposed to optimize cost and average waiting time of heterogeneous tasks. The uniqueness of the proposed model is that it makes task execution schedule and sends calculated data to resource manager before actual task execution for reducing average waiting time of tasks. Further, we validate the effectiveness of the proposed model using GridSim by considering random values of task and resources. Experimental results show that the proposed model provides up to 37% reduction in average waiting time than existing approaches under various environment conditions.

Keywords: Grid computing, Resource allocation mechanism, Cost matrix, Scheduling, Waiting time.

1. Introduction

In a grid environment, various resources are geographically distributed and used for meeting the demand of high-performance computing. Due to dynamicity and availability issue, discovery [1], selection and allocation of resource [2] is challenging task. Broadly every resource allocation technique follows three phases [3]-

- Resource discovery- In this phase, a list is made of all resources which are accessible for executing the user request.
- Mapping- Before real execution of the assignment, a mapping between resource and task is done according to the need of a user.
- Execution- This is the last phase in which a resource is assigned to a task for execution.

Various resource allocation algorithms are used for executing tasks submitted by user. The determination of resource relies upon various criteria like cost, time, and other Quality of Service (QoS) metrics. Most of the literature of resource allocation techniques is done by optimizing metrics like- resource utilization, execution time, cost, schedule length. Various issues have been found when a task is assigned to a resource in grid environment-

- A resource may join or leave the network at any time due to dynamic nature of resources
- More than one task may be assigned to a resource due to cost or deadline constraint associated with task

In this paper, to solve the scheduling issue, we consider workload of multiple tasks which select the same resource and propose a resource allocation model for web server Grid. Our major contributions are as follows-

(1) Firstly, a resource selection mechanism which selects least cost resource for tasks to reduce Total Execution Cost (TEC)

(2) Secondly, a schedule mechanism for tasks which selects the same resource due to least cost constraint to reduce Average Waiting Time (AWT)

The rest of the paper is organized as follows. In section 2, some resource allocation algorithms have been analyzed and categorized on the basis of technology, technique, strength, limitations and future scope of work. The section 3 describes proposed resource allocation model and algorithm, section 4 elaborates results and analysis. The final section includes conclusion and future work.

2. Related work

Various resource allocation algorithms have been already exists with cost and time constraints. Murugesan and Chellappan 2011 [4] introduced a resource allocation model which considers QoS requirement for fulfilling the user request.

An auction method is used by Izakian et al. 2010 [5] to submit a task to a suitable resource that results in minimum execution cost. Further, work can be improved by adding budget and deadline constraint with task. Resources and users acts as provider and consumer agents respectively. Provider agent takes decision on the basis of its workload, and consumer agent takes decision on the basis of two constraints- time and resource remaining respectively. The presented method improves resource utilization and reduced execution rate.

Shah et al. 2010 [6] analyzed least cost method and presented a modified least cost method by dividing task into sub tasks on various resources for reducing the execution cost. Scheduler considers the least cost criteria for selecting the resource.

A mathematical model is obtained for modified Vogel Approximation Method (VAM) by introducing an index Singhal et al. 2013 in [7]. A matrix is used for base utilization price of resource for execution of task. In presented model, scheduler calculates the difference between lowest and next lowest base price respectively. Two base price differences are calculated corresponding to each row and column respectively. A difference set is formed by joining two sets. Scheduler finds an index value which is used to map a task to a resource. Results are compared with different resource allocation strategies (Shah et al. 2010 [6], Kamalam and Bhaskaran 2011 [8], Korukoglu and Balli 2011 [9]) and gives better result in each scenario.

A cloud consist various heterogeneous resources to meet the user demand. When an algorithm is designed, it must ensure the optimum use of each resource. Resource state is analyzed and a model is introduced by Nie et al. 2017 [10] to make the resource fully utilized. Resource is assigned based on resource state and energy consumption of physical machines. The algorithm discussed by Nie et al. 2017 [10] provides less energy consumption and service level agreement violations than First-Fit - Minimum Migration Time algorithm and Power Aware BFD-MMT algorithm.

The main aim of service provider is to gain revenue, whereas aim of end user is to reduce cost. Dynamic architecture of cloud computing is analyzed by Weintraub and Cohen 2015 [11]. Three price cost minimization models are also presented for customer point of view - hierarchal, simple pricing, complete pricing model respectively. The model can be extended in future by adding consumer and provider preference in cost models.

A workflow task scheduling algorithm is presented by Kaura and Singh 2016 [12] to reduce the execution time based on user defined budget limit. Tasks are scheduled to reduced execution time & cost with increased reliability. Presented algorithm shows better result than Basic Randomized Evolutionary Algorithm (BREA) in terms of cost & execution time.

A task scheduling algorithm is presented by Chena et al. 2017 [13] to meet two objectives- (1) to meet the budget defined by user (2) to minimize the scheduled length. Schedule length is minimized by using heuristically scheduling with low time complexity. Work is compared with HBCS Arabnejad and Barbosa 2014 [14] DBCS Arabnejad et al. 2016 [15] to show the improvement using presented approach.

To meet the market demand from the user and service provider perspective, a dynamic pricing scheme is introduced Shaari et al. 2017 [16] for resource allocation in cloud environment. Pricing function is calculated based on utilization of resources. It promotes the low utilized resources and discouraged over utilized resources. Cost is analyzed and compared with Round Robin and Random allocation algorithms for data intensive jobs. For both kinds of jobs, described scheme provides low cost than Round Robin & Random allocation algorithms.

An optimized mechanism for resource allocation is presented by Singh and Kumar 2015 [17]. Tasks are submitted by user with their workload. By checking resource availability, a least cost resource is selected. User submits task in task pool where tasks reside for getting the resource. Scheduler is responsible for task allocation to a resource. But before assignment of tasks, Resource Cost Monitor

Table 1. Survey of resource allocation techniques

Author	Technique	Strength	Limitation & Future Scope
Murugesan & Chellappan [4]	Resource allocation	Cost	Budget & time constraint
Izakian et al. [5]	Auction based resource allocation	Execution rate, resource utilization	Not given
Shah et al. [6]	Cost based resource allocation	Cost	Performance optimization with same cost
Singhal et al. [7]	Parallel execution based resource allocation	Cost	Virtual simulation on Cloudsim
Nie et al. [17]	Energy based resource allocation	Energy consumption	Real world experiment
Weintraub & Cohen [11]	Cost optimization	Cost	Consumer & provider preference
Kaura & Singh [12]	Budget & time constraint scheduling	Reliability, cost, execution time	Comparison with other scheduling algorithms
Chena et al. [13]	Budget constraint based	Budget constraint & task scheduling	Cost aware design schedule length
Shaari et al. [16]	Price based resource allocation	Cost	Power consumption & user feedback
Singh & Kumar [17]	Cost based resource allocation	Cost	Time constraint
Muthu & Enoch [18]	Resource scheduling	Throughput, resource utilization	Simulation for workflow models
Loganathan et al. [19]	Energy aware job scheduling	Energy consumption	Simulation on federated cloud environment
Nagaraju & Saritha [20]	Resource scheduling	Makespan, queue length	Fault tolerance mechanism

(RCM) searches for min cost resource and sends the information to scheduler. After getting information from RCM, scheduler assign task to resource and kept in allocated resource list. The algorithm gives the lower execution cost in comparison to algorithm introduced by Singhal et al. 2013 [7].

An optimized mechanism for resource allocation is presented by Singh and Kumar 2015 [17]. Tasks are submitted by user with their workload. By checking resource availability, a least cost resource is selected. User submits task in task pool where tasks reside for getting the resource. Scheduler is responsible for task allocation to a resource. The algorithm gives the lower execution cost in comparison to algorithm introduced by Singhal et al. 2013 [7].

A reliable hybrid model of Genetic Algorithm (GA) and Bacterial Foraging Optimization Algorithm (BFOA) is proposed by [1] and [18]. The tasks are assigned to suitable resource to minimize makespan, cost and throughput. The algorithm can be enhanced for workflow models. A self learning algorithm can be implemented for best resource selection.

A task scheduling mechanism is presented by [3, 19] to optimize the energy consumption in cloud environment. The energy consumption is reduced by considering the pre-emption policy of jobs and by making less number of hosts in active state. The result shows better performance for makespan, throughput and success rate also.

A genetic algorithm based resource scheduling is presented for mobile cloud computing by [2, 20]. The crossover and mutation is used to generate new population. The simulation result shows the reduction in data transfer time, execution time, queue length and provides maximum resource utilization than other optimization methods.

Various conventional techniques are analyzed in Table 1 and classified on the basis of technology, strength, drawback and future scope. Most of the research work focused on resource allocation by considering only resource characteristics such as cost, energy, deadline, etc. We proposed a resource allocation model by considering resource as well as task characteristic to optimize the average waiting time of tasks. Cost and workload are considered as a resource and task characteristics respectively.

3. Proposed resource allocation model

Fig. 1 shows the proposed resource allocation model by considering workload and resource availability of tasks and resources respectively. The proposed mechanism ensures selection of least cost

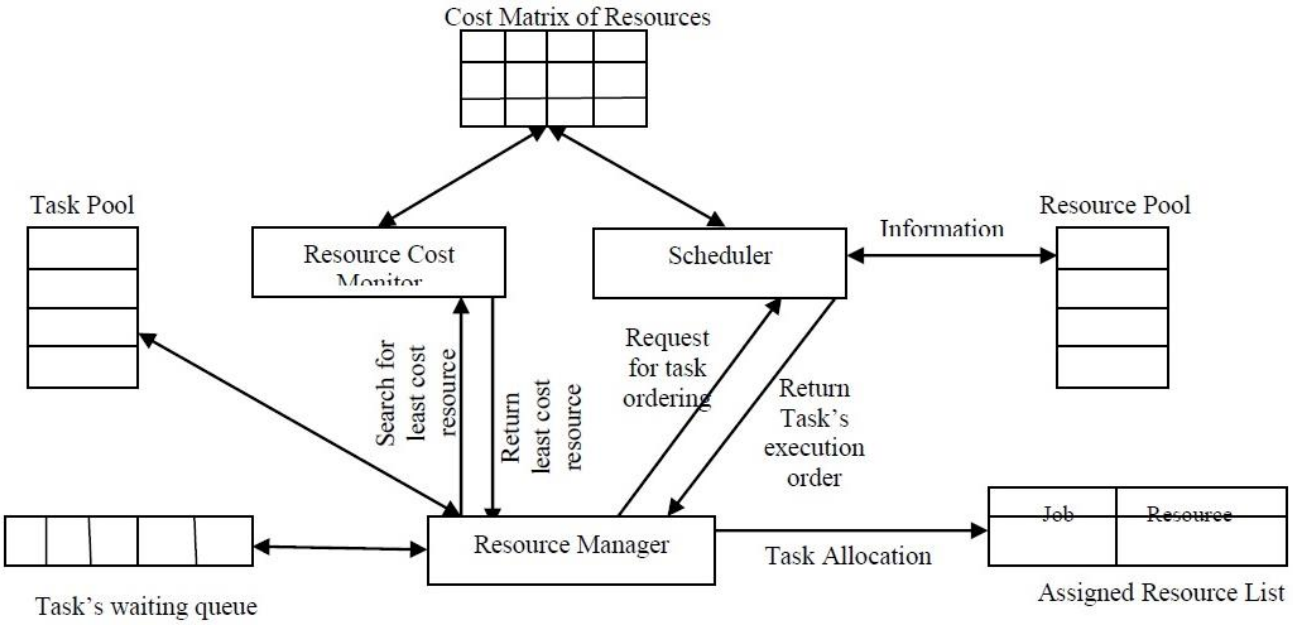


Figure. 1 Proposed resource allocation model

resource with minimum average waiting time, which improves the overall performance of the system.

Various tasks join the task pool to execute on suitable resource. Resource Manager (RM) is responsible for actual execution of task to a resource. RM selects a task from task pool where all tasks are waiting for assignment of resource. The Resource Cost Monitor (RCM) search the least cost resource from cost matrix and sent to the RM. Availability of resource is checked by comparing with task workload. If resource availability is larger than task's workload, then task can execute on selected resource, otherwise next least cost resource is searched for execution. The same process is repeated until task's pool doesn't become empty.

But due to least cost constraint, many tasks may select the same resource for execution. But at a time only one task can take service from a resource. So ordering is required of tasks to decide which will go first for execution. The major responsibility of scheduler is to make schedule of tasks which selects the same resource. For ordering of tasks, shortest workload first is implemented, Task having less workload will execute first. Scheduler compares the workload of tasks and arranges in ascending order of workloads. Tasks which select different resource do not require ordering in execution. Ordered list is sent to RM where actual task allocation is done and execution takes place.

3.1 Proposed resource allocation algorithm

The main objective of the proposed algorithm is that if a resource is selected by more than one task then task having least workload will execute first to reduce AWT of queued jobs. The algorithm chooses the least cost resource for task execution by checking resource availability. It also generates a schedule for task execution based on shortest workload to reduce AWT. The algorithm assigns a resource for task execution and determines Total Execution Time (TEC) and AWT for all tasks.

The step 1-3 of the algorithm determines the content of Cost Matrix (Cij), Workload (WLD), and Resource Availability (RA). The Resource Manager (RM) chooses the least cost resource in step 7. Further, resource availability is checked by comparing with WLD and task is added to resource queue in step 9. If tasks already exist in resource queue, scheduler arranges tasks in incremental order of workload in step 11 and computes task waiting time. If there is only one resource in resource queue, the task is executed on the same resource without any delay. After each resource allocation, Resource Availability Remaining (RAR) is computed using Eq. (1) in step 22. The algorithm also considers the dynamic nature of tasks and resources in the grid environment. In step 23-26, cost, TEC and AWT are computed using Eq. (2), (3) and (4) in step 23-26, where Total Waiting Time (TWT) is the waiting time of all the jobs residing in job pool.

$$RAR=WLD-RA \quad (1)$$

$$Cost\ (C) = Least\_cost \times WLD \quad (2)$$

$$TEC = \sum_{j=1}^{m} Cij \times WLD \quad (3)$$

$$AWT = TWT / n \quad (4)$$

Algorithm: Resource Allocation Algorithm

Input: Tasks (T1, T2....Tn), Cij, WLD, RA
Output: TEC, AWT

BEGIN
1. **for all tasks do**
2. Determine content of Cij, WLD, RA
3. **end for**
4. **for each task do**
5. **while** task exists **do**
6. **for all resource do**
7. //Check the availability of selected resource
8. if Ti (WLD)<=RA {
9. Submit job in $Resource_i$_queue
10. **if** $Resource_i$_queue>1{
11. Scheduler arranges tasks in increment order of workload
12. goto step 26 to compute waiting time
13. **endif**
14. **elseif** Resource_queue = =1
15. Only one task selects the resource and goes for execution
16. **else**
17. No task is assigned to resource
18. **end for**
19. **end while**
20. **end for**
21. **for all task do**
22. Calculate RAR using Eq. (1)
23. Calculated C using Eq. (2)
24. Calculated TEC using Eq.(3)
25. **end for**
26. Calculate AWT using Eq. (4)
27. Send calculated data TEC, AWT to RM

END

4. Simulation and results

For the simulation of proposed resource allocation model, GridSim toolkit is used and results are compared with existing approaches [17] [20]. The results are compared for the reduction of AWT of tasks. For simulation, we use system with windows 7, Intel(R) Pentium(R) CPU B940 @ 2.00 GHz 2.00 GHz. 2.00 GB RAM configuration. The simulation is performed for various cases (task = resource, task > resource, task < resource).

4.1 Modules of proposed resource allocation algorithm

As number of resources gets costly for organizations, efficient scheduling policy is required to optimize the cost of resources as per task requirement. Here, Cost Matrix is used for deciding least cost resource based on resource availability. Another key issue is waiting time of tasks for getting the resource. For reducing average waiting time, tasks are arranged in incremental order of workloads. Tasks which select the same resource require ordering in such a manner so waiting time of tasks may reduce. The proposed algorithm is further distributed in various modules and later all modules are integrated to form proposed resource allocation model. The modules are as follows-

- Establishing the architecture involving tasks, resources, workload of tasks (WLD) , and Resource Availability (RA)
- Calculation of least cost for all the tasks to respective resources
- Calculation of AWT by scheduler
- Sending all valid calculated data to RM which would act as a decisive system and decide best suited resource for task.

Cost is measured in dollar ($), workload and waiting time are measured in milliseconds (ms). Following methods are used for implementing various modules of proposed resource allocation algorithm –

- set_column () & set_row()- To enter number of tasks and resources
- getdata()- To enter content of Cost Matrix
- getworkload()- To enter workload of each task
- getRA()- To enter availability of each resource
- make_allocation()- For allocating task to resource by using getMinVal(temp_data) method, which identify least cost for the task
- Avg_waiting_time()- For calculating average waiting time of tasks. First task in queue has waiting time 0.
- display()- Show the calculated least cost and AWT of tasks.

Ranges of various parameters are given in Table 2 which is used in simulation.

Table 2. Simulation parameters

Parameter	Range
Number of Tasks	4-100
Number of Resources	5-100
Number of iterations	10 for each case
Workload	5-500 ms
Resource cost	2-50 $

4.2 Performance evaluation

For making proposed model more generic, we use Eq. (5) to get simulation results for three different cases –

- Case 1- Simulation with Tasks (T) > Resource (R) (0<m<1)
- Case 2- Simulation with Tasks < Resource (m>1)
- Case 3- Simulation with Tasks = Resource (m=1)

$$Task\ (T) = m \times Resource\ (R) \quad (5)$$

Where m is random value which is greater than 0.On the basis of value of m, three cases are formed-

4.2.1. Case 1- Simulation with Tasks > Resources (0<m<1)

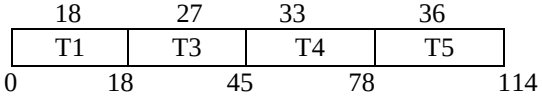
In this case, the simulation is performed assuming numbers of tasks are greater than resources. In following example, If T1 selects resource R1 for execution, then R1 will charge 7 $ per ms. If T2 selects resource R2, then R2 will charge 5 $ per ms and so on. In this example, task T1, T3, T4, and T5 select the same resource R1, But only one task can use a resource at a time. According to proposed algorithm, shortest workload task is selected first for execution which will reduce the average waiting time of tasks. Table 3 shows selection of least cost resources by considering resource availability and workload of task [7]. TEC is calculated as follows-

TEC=(7×18)+(5×24)+(3×27)+(22×33)+(15×36)+(13×30) = 1983 $

Table 3. Allocation of tasks

T/R	R1	R2	R3	R4	R5	WLD
T1	7(18)	7	11	12	11	18
T2	11	5(24)	14	8	9	24
T3	3(27)	7	11	13	17	27
T4	22(33)	3	12	6	11	33
T5	15(36)	7	8	9	12	36
T6	11	7	14	13(30)	14	30
RA	45	25	30	30	25	

Table 4. Gantt chart of tasks execution



- Analysis of Average Waiting Time

We are assuming that all the Tasks are arriving at time 0. Computation of AWT of tasks according to proposed algorithm is shown in Table 4.

The Fig. 2 shows the AWT of Existing approaches [17, 20] and proposed approach versus number of tasks. The number of tasks varies from 6-100. The AWT is the time difference between task arrival and execution start time. From the simulation, it is analyzed that proposed model reduces up to 31% of the AWT over other algorithms. The proposed model provides 108 ms AWT when number of tasks is 20 rather than 158 ms at the other algorithms. The reduction in AWT is due to workload based scheduling of tasks before actual execution.

4.2.2. Case 2- Simulation with Tasks < Resources (m>1)

In this case, the simulation is conducted assuming numbers of tasks are less than resources. Table 5 shows that Tasks T1, T2, T3, and T4 select resource R3 and will take 6, 9, 6 and 8 ms respectively. TEC is calculated as follows-

TEC= (2×6) + (2×9) + (2×6) + (2×8) =58 $

- Analysis of Average Waiting Time

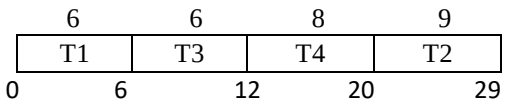
Tasks T1, T2, T3, and T4 select resource R1 for execution and have workloads 6, 9, 6 and 8 ms respectively. AWT of tasks is shown in Table 6.

The Fig. 3 shows the AWT of Existing approaches and proposed approach versus number of tasks. The number of tasks varies from 4-100. The result shows that proposed model reduces up to 37% of the AWT over other algorithms. The proposed model provides 80 ms AWT when number of tasks is 60 rather than 127 ms at the other algorithms.

Table 5. Allocation of tasks

T/R	R1	R2	R3	R4	R5	WLD
T1	8	9	2(6)	8	8	6
T2	8	9	2(9)	8	8	9
T3	8	9	2(6)	8	8	6
T4	8	9	2(8)	8	8	8
RA	12	12	12	12	12	

Table 6. Gantt chart of tasks execution



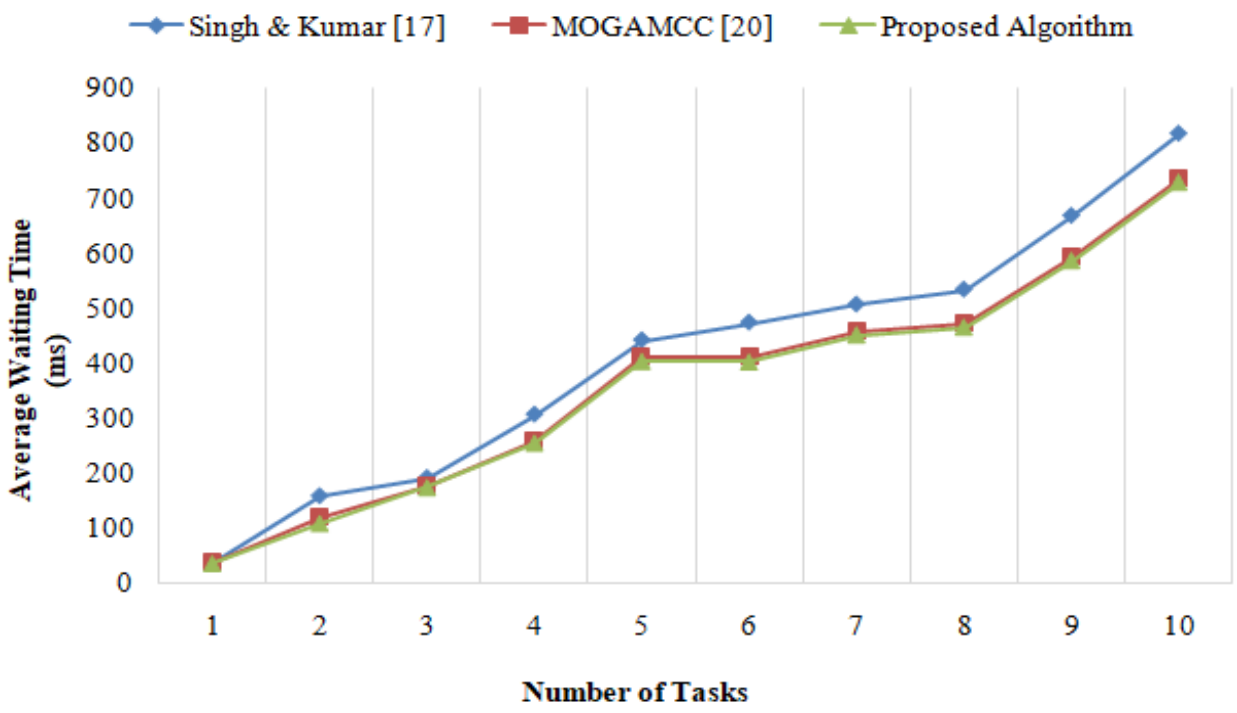


Figure.2 Average Waiting Time of Tasks (> Resource**)**

4.2.3. Case 3- Simulation with Tasks = Resources (m=1)

In this case, the simulation is conducted assuming number of tasks and resources are equal. Table 7 shows selection of least cost resources by considering RA and workload of task [21]. TEC is calculated as follows-

$$TEC=(5\times28)+(5\times35)+(8\times33)+(8\times45)+(5\times35)+(5\times40)+5\times30)+(5\times39)=1659\ \$$$

- Analysis of Average Waiting Time

The AWT of tasks according to proposed algorithms is shown in Tables 8, 9, and 10.

Table 8. Gantt chart of tasks execution-step I

	28	30	
	T1	T7	
0		28	58

Table 9. Gantt chart of tasks execution-step II

	35	39	45	
	T5	T8	T4	
0		35	74	119

Table 10. Gantt chart of tasks execution-step III

	33	35	
	T3	T2	
0		33	68

Table 7. Allocation of tasks

T/R	R1	R2	R3	R4	R5	R6	R7	R8	WLD
T1	5(28)	11	15	13	7	6	12	16	28
T2	17	20	16	19	15	16	14	5(35)	35
T3	11	9	12	8	12	17	20	8(33)	33
T4	9	11	8(45)	9	13	19	5	21	45
T5	9	8	5(35)	15	12	17	8	15	35
T6	8	9	16	16	5(40)	8	12	17	40
T7	5(30)	13	17	9	19	13	7	9	30
T8	11	6	5(39)	11	20	17	14	6	39
RA	40	50	50	25	50	45	30	50	

The Fig. 4 shows the AWT of Existing approaches and proposed approach versus number of tasks. The number of tasks varies from 4-100. From the simulation, it is analyzed that proposed model reduces up to 30% of the AWT over other algorithms. The proposed model provides 35 ms AWT when number of tasks is 60 rather than 50 ms at the other algorithms.

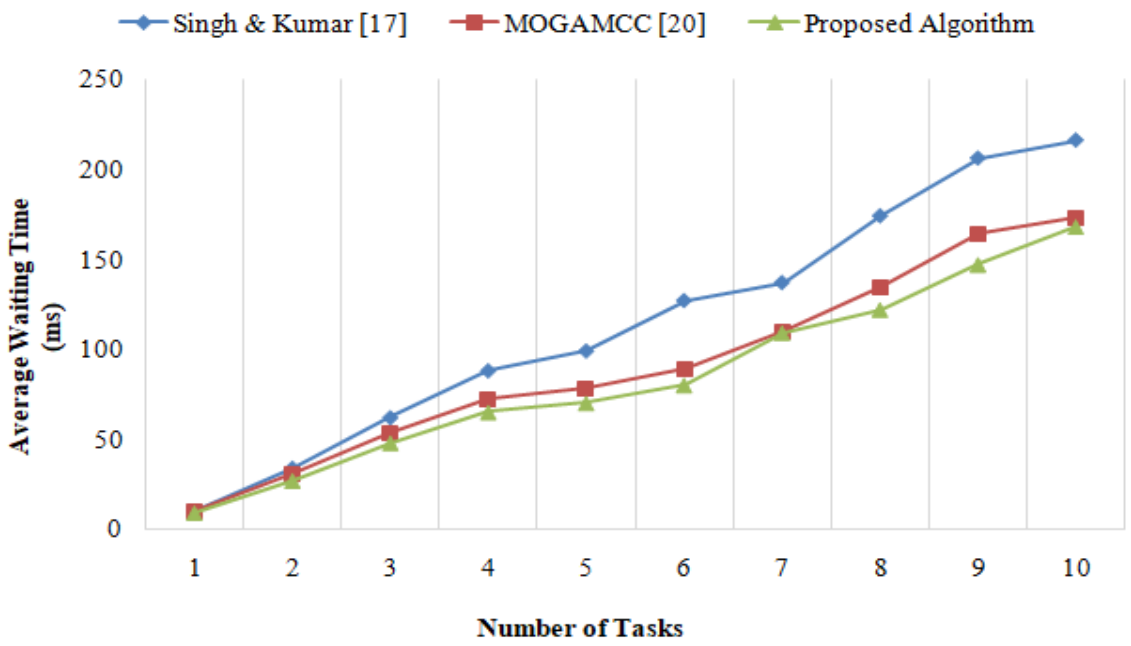


Figure.3 Average Waiting Time of Tasks (< Resource)

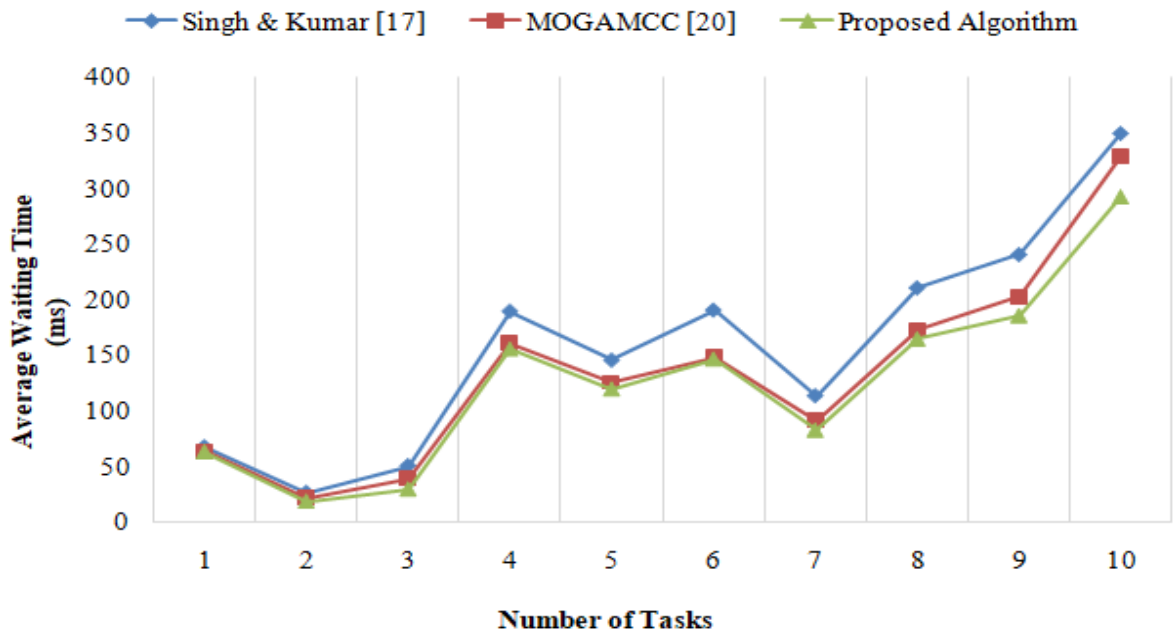


Figure.4 Average Waiting Time of Tasks (= Resource)

Table. 11 Comparison between proposed and existing approaches

Criteria	Singh & Kumar [17]	MOGAMCC [20]	Proposed Model
Technique	Resource selection mechanism	Task scheduling mechanism	Resource selection & task scheduling
Task sorting basis	Least cost	Not performed	Least cost & task workload
Scheduling	No	Yes	Yes
AWT	High	Average	Low
% reduction in AWT	-	up to 37.01	up to 12.5

In each case, there is noticeable reduction in AWT while considering least workload as criteria instead of least cost for ordering of jobs. Responsibility of RM is to check the state of allocated resource consistently. When it completes the current executing task, resource is assigned to next task which is waiting for it. If the resource doesn't fulfil the requirement of task, then next least cost resource is searched.

In all the cases AWT of our proposed approach is found to be less than existing approaches [17, 20]. The reason behind is the effective resource selection and task scheduling approach on the basis of cost and task workload respectively. Table 11 shows the comparative analysis between proposed and existing approaches based on various parameters.

5. Conclusion and future scope

In this paper, analysis of existing resource allocation technique is carried out and classification is done based on various parameters. Based on literature, various challenges of grid computing are analyzed like- cost, scheduling, load balancing etc. Due to least cost constraint, sometimes more than one task selects the same resource. In that scenario, tasks need to be ordered before actual execution. We have proposed a resource allocation model that provides mapping between tasks & resources with least cost and minimum AWT. The uniqueness of this model is that it makes schedule of tasks and sends calculated data to RM before actual execution for reducing waiting time of tasks for resource. The proposed algorithm worked well in all cases (T>R, T<R, T=R).We have compared proposed model with existing approaches Singh and Kumar [17] and MOGAMCC [20] and observed up to 37 % and 12.5 % reduction in AWT respectively.

The proposed work can be further extended to achieve following objectives in future-

- Mechanism to optimize other metrics like response time, penalty ratio, time constraint
- Implementation of fault tolerance mechanism and perform simulation in real time environment.

References

[1] W. Depoorter, K. Vanmechelen, and J. Broeckhove, "Advance reservation, co-allocation and pricing of network and computational resources in grids", *Future Generation Computer Systems*, Vol.41, pp. 1-15, 2014.

[2] A. Yarygina and B. Novikov, "Optimizing Resource Allocation for Approximate Real-Time Query Processing", *Computer Science and Information Systems*, Vol. 11, No. 1, pp. 69–88, 2014.

[3] F. Dong and S. G. Akl, "Scheduling Algorithms for Grid Computing: State of the Art and Open Problems", *Technical Report No. 2006-504, Queen's University*, Canada, 2006.

[4] G. Murugesan and C. Chellappan, "An Economic Allocation of Resources for Divisible Workloads in Grid Computing Paradigm", *European Journal of Scientific Research*, Vol. 65, No. 3, pp. 434-443, 2011.

[5] H. Izakian, A. Abraham, and B. T. Ladani, "An auction method for resource allocation in computational grids", *Future Generation Computer Systems*, Vol. 26, No. 2, pp.228-235, 2010.

[6] S. N. M. Shah, A. K. B. Mahmood, and A. Oxley, "Modified Least Cost Method for Grid Resource Allocation", In: *Proc. of International Conf. on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pp.218-225, 2010.

[7] S. Singhal, M. Kumar, and K. Kant, "An Economic Allocation of Resources in Grid Environment", In: *Proc. of IEEE International Conf. on Information Systems and Computer Networks*, pp. 185-190, 2013.

[8] G. K. Kamalam and V. M. Bhaskaran, "An Efficient Hybrid Job Scheduling Algorithm for Computational Grids", In: *Proc. of International Conf. on Web Services Computing*, pp. 6-10, 2011

[9] S. Korukoglu and S. Balli, "An Improved Vogel's Approximation Method for The Transportation Problem", *Mathematical and Computational Applications*, Vol. 16, No. 2, pp. 370-381, 2011.

[10] J. Nie, J. Luo, and L. Yin, "Energy-aware Multi-dimensional Resource Allocation Algorithm in Cloud Data Center", *KSII Transactions on Internet and Information Systems*, Vol. 11, No. 9, 2017.

[11] E. Weintraub and Y. Cohen, "Cost Optimization of Cloud Computing Services in a Networked Environment", *International Journal of Advanced Computer Science and Applications*, Vol. 6, No. 4, 2015.

[12] N. Kaura and S. Singh, "A Budget-Constrained Time and Reliability Optimization BAT Algorithm for Scheduling Workflow Applications in Clouds", In: *Proc. of the 7th International Conf, on Emerging Ubiquitous Systems and Pervasive Networks*, pp.199 – 204, 2016.

[13] W. Chena, G. Xiea, R. Lia, Y. Baia, C. Fana, and K. Lia, "Efficient Task Scheduling for Budget Constrained Parallel Applications on Heterogeneous Cloud Computing Systems", *Future Generation Computer Systems*, Vol. 74, pp. 1-11, 2017.

[14] H. Arabnejad and J. G. Barbosa, "A budget constrained scheduling algorithm for workflow applications", *Journal of Grid Computing*, Vol. 12, No. 4, pp. 665-679, 2014.

[15] H. Arabnejad, J. G. Barbosa, and R. Prodan, "Low-time complexity budget-deadline constrained workflow scheduling on heterogeneous resources", *Future Generation Computer Systems*, Vol. 55, pp. 29-40, 2016.

[16] N. A. B. M. Shaari, T. F. Ang, L. Y. Por, C. S. Liew, "Dynamic Pricing Scheme for Resource Allocation in Multi-Cloud Environment", *Malaysian Journal of Computer Science*, Vol. 30, No. 1, 2017.

[17] H. Singh and S. Kumar, "Optimized Resource Allocation Mechanism for Web Server Grid", In: *Proc. of IEEE UP Section International Conf. on Electrical Computers and Electronics*, pp. 1-6, 2015.

[18] A. B. A. Muthu, and S. Enoch, "Optimized Scheduling and Resource Allocation Using Evolutionary Algorithms in Cloud Environment", *International Journal of Intelligent Engineering and Systems*, Vol. 10, No. 5, pp. 125-133, 2017.

[19] S. Loganathan, R. D. Saravanan, and S. Mukherjee, "Energy Aware Resource Management and Job Scheduling in Cloud Datacenter", *International Journal of Intelligent Engineering and Systems*, Vol. 10, No. 4, pp. 175-184, 2017.

[20] D. Nagaraju and V. Saritha, "An Evolutionary Multi-Objective Approach for Resource Scheduling in Mobile Cloud Computing", *International Journal of Intelligent Engineering and Systems*, Vol. 10, No. 1, pp. 12-21, 2016.

[21] K. Kumar, S. Singhal, and S. P. Tripathi, "A Resource Allocation Algorithm for Heterogeneous Jobs in Grid Environment", In: *Proc. of IEEE International Conf. on Recent Advances and Innovations in Engineering*, pp. 1-6, 2014.