

# Resource and Task Allocation Scheduling in Distributing System for Optimizing Execution Time

Ms. Ruchi Arora<sup>1</sup>, Mr. Anuj Mehta<sup>2</sup>

<sup>1</sup>Mtech Student, SKIET, Kurukshetra

<sup>2</sup>Assistant Professor, Department of Computer Science, SKIET Kurukshetra

\*\*\*

**Abstract** - Grid computing consists of achieving an effectual clustering of the valuable resources having dissimilar locations which will deal with real time scenarios. The grid follows the dispersed procedures having heavy workloads which can be in the form of the traffic files from different locations. Grid computing is related to the extraordinary performance systems like computer clustering or we can say nodes in the grid in such a manner that each set of the node performs different tasks and applications. Grid computers also deals with networks with topology variations and diverse geography which is not essentially to connect substantially to the cluster of computers. As the number of traffic increases day by day, is the challenging task to complete all the allocated processes in the limited time intervals. So this research deals with the efficient scheduling and optimization approach for the resource management using Ant colony optimization and round robin scheduling to obtain low execution intervals with less error rate probabilities. The whole simulation is done in GRIDSIM environment.

**Key Words:** Distributed System, Grid Computing, Resource Management, Optimization, Task Scheduling

## 1. INTRODUCTION

The progression of distributed computing which is popularly known as the grid computing deals with massive opportunities for the governments and establishments to use various processing cycles using computers networks is one of the main key structure in the world of working servers [1,2]. The knowledge of grid computing is associated to the peer to peer communication or computing model structure which is having great demand in real time applications. The inventor of the grid or distributed computing uses a comparable prototype to gather the resources accessible in detached clusters of desktop processors or servers [3, 4]. Grid computing raises these groups or clusters to the high level by linking multiple collections over geologically distributed areas for improved group effort and sharing of various valuable resources [10]. The initial form of grid computing initiated with the perception of circulated computing. Today, the grid computing is observed as the modern generation based IT infrastructure which is expected to convert complex computation and collaborations to the real time working web servers. The gathering of networks will occur, each inside its own background which is shared by various high communities in the similar industry with high rate of interests. Grids are service driven over the internet for the need of utilities or resources required by the users [11, 12].

Analysts approximates that extensive implementation of grid computing takes various phases to be implemented at high scale. The very first criteria involves the profitable operation of construction grids inside major firms which are having global occurrence or a requirement to assessment of the resources separately in a solitary corporate site [13]. The second part or phase deals with the Partner's grid which will occur as administrations functioning within comparable productions and areas of concentration which rely on real time projects, and use of neighbour resources to reach a shared objective. For instance, organizations based on life sciences functioning together might quicken their research period by binding the available computer systems or by using large amounts of information sharing within partner administrations [14, 15]. The third and the very importance phase is the service grid which occurs as operators implement the grid like there source model. It is in the same way like consumers are very self-assured in their usage of the internet; prevalent acknowledgment of the assistances of grid computing will yield it to the subsequent level [16, 17].

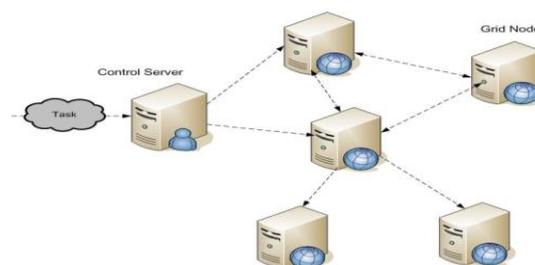


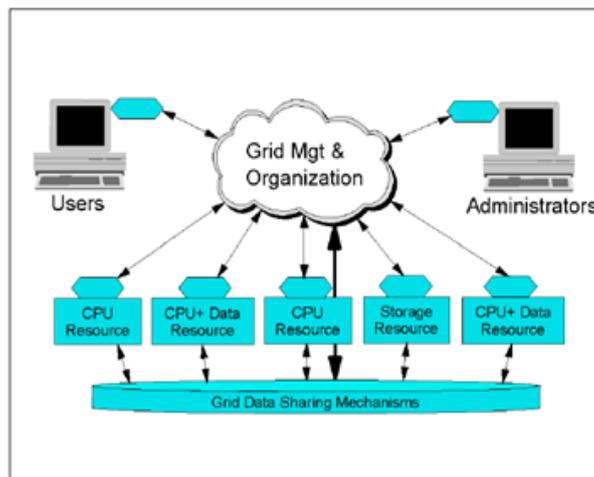
Fig. 1: Grid Computing Architecture

The idea of grid computing is certainly equivalent or dispersed computing but the variance is in the part of scaling and the complexity of the structure. Instead of distribution of one or extra resources or services for parallel handling of the systems, every computing source is mutual collaboration of all the mainframes in the system [18, 19].

Grid computing associates with the idle handing out control of numerous computing components and operates that handing out power to calculate one process which is further divided into various jobs. The jobs can be the storage of the data or can be difficult computations. The grid contains several diverse approved mixed schemes and possibly be possessed by numerous other organizations[20, 21].

**Grid computing assistances**

1. Grid computing deals with the organizations which provide collective resources in the entire IT arrangement and it doesn't matter where the systems are situated. It rejects circumstances wherever one location is successively working on extreme volume, while other systems have sequences to standby [22].
2. Organizations can recover the superiority and swiftness of the goods and facilities they transport, while dropping IT charges by empowering translucent relationship and sharing of the resources [23].
3. Grid computing deals with the widely spread administrations to easily cooperate on developments by generating the capability to share the resources from software presentations and data to the practical applications.
4. Grid computing can generate a more healthy and strong IT substructure which is able to rejoin to slight or major tragedies [24, 25].



**Fig. 2:** Grid Computing Environment

**2. RELATED WORKS**

There are various valuable studies on study of grid resource distribution in various distributing arrangement. For an active resource distribution procedure, it is intolerable to alter the state material regularly because of the high overhead of the communications between the nodes (also termed as connected computers). Manpreet Singh [5] proposed Grid Resource Allocation concept and is proficient of achieving high quality provision of grid resources to the users and also they have computed the cost and execution time as their performance parameters. Harshad kumar B. Prajapati, Vipul A. Shah et al. [6] proposed perceptions of scheduling of resources and also discuss the arrangement of scheduling procedures. Also, they have discussed the methodology for evaluating scheduling processes comprising both real time scheme and replication based methods. Minakshi Memoria, Dr. Mukesh Yadav et al. [7] recover the resource operation with supreme effectiveness and quantity even in existence of liability in structure. It comprises three segments analysis of job, allocation of resource phase, and executions of the job. It deals with the amount of allocations with the abilities that decrease likelihoods of allocation of the resources which are having more probability of the failure among the various processors, their memories and the processors. Mr. Kapil B. Morey, Prof. Sachin B. Jadhav et al. [8] presented an efficient approach for the balancing of the dynamic load using grid environment which achieves high recital computation using optimal geographically system and heterogeneous allocation of the resources. They have worked on the diminishing of the load using load balancing technique to reduce the overhead of the

system. Rab Nawaz, WuYang Zhou, Muhammad Usman Shahid, Osman Khalid et al. [9] have worked on the grid computing libraries and also worked on the middleware layer which deals with the division of the jobs or tasks and distribution among the several computing resources

### 3. PROPOSED MODEL

In the proposed technique we have used hybridization of the two algorithms which are named as the round robin process and the optimization algorithm named as Ant Colony optimization.

The round robin arrangement is a very effective process scheduling and also acts as a scheduler in the field of computing. In scheduling process we have make efficient use of time quanta which is assigned the each job or process in handling the execution of the user request in a circular fashion on the basis of priority which will further reduce the overhead in the queuing process. Its execution is very simple and the scheduling through RR is able achieve better data packet handling in resource allocations in the grid computing. Further we have used ant colony optimization in hybridization with round robin to optimize the performance of the grid computing by using intelligent searching in terms of high load which balances the overhead and decreases the cost and execution time for the efficient processing in the grid computing.

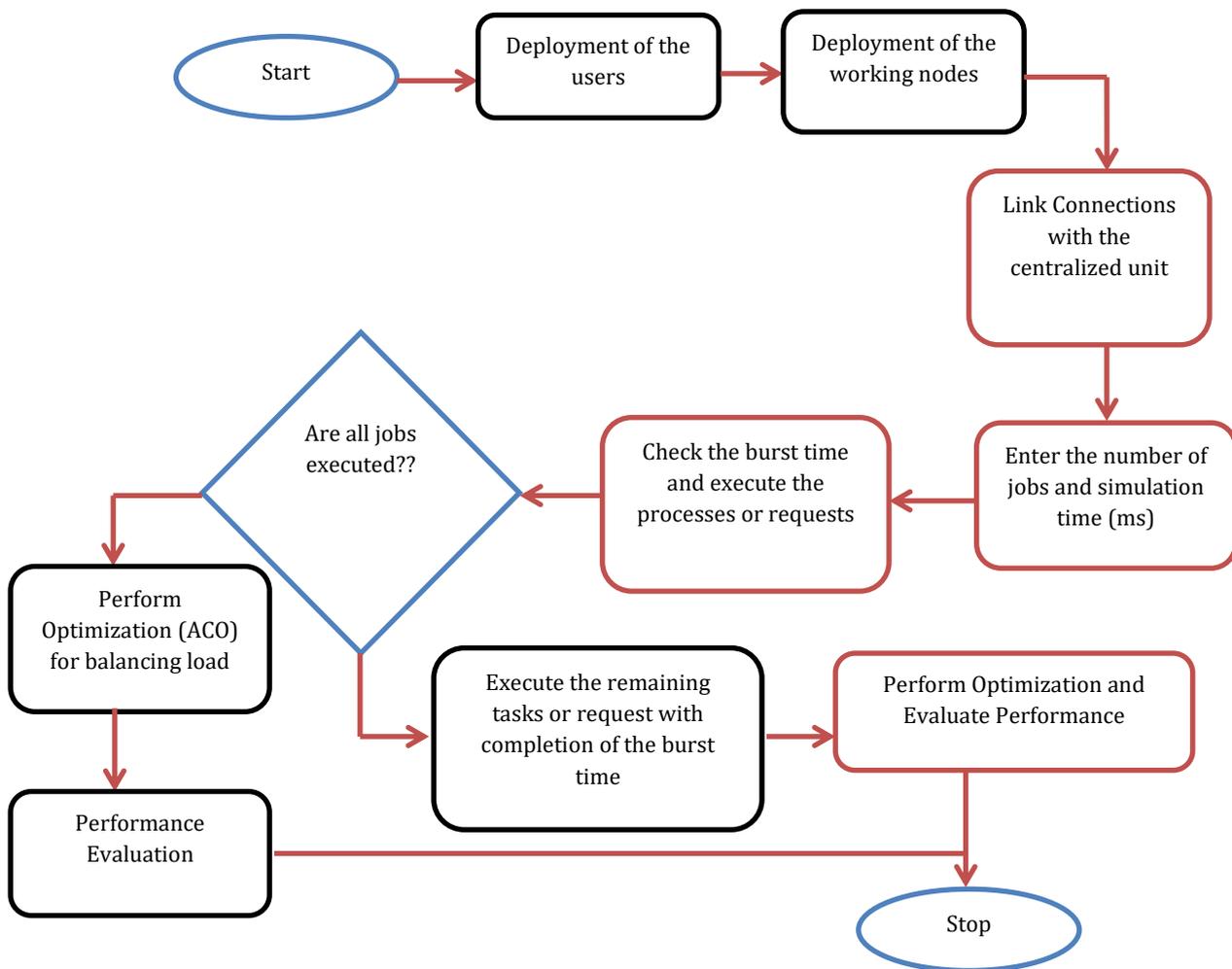


Fig. 3: Proposed Flow Diagram

In figure 3 the proposed model is explained in which we have designed the scenario for minimum execution time and less cost. Initially we have entered the number of the user for the deployment and then we have entered the number of resource to be allocated. Then the deployment of the center nodes and also the connected nodes will be attained to make the grid computing scenario. After grid environment we have worked on the scheduling of the tasks. If the process takes the time which is less than

the allocated time then the process will be completely executed in the first time otherwise it will go to the pending state and then the process will be evaluated using round robin again for the incomplete jobs with the remaining allocated time.

So the processing takes the lot of load to be executed and for that the optimization process using Ant colony optimization is evaluated to optimize the load by reducing the standard deviations of the processes or jobs to equalize the load and to achieve less complexity, less latency and less cost of the system and the allocation of the resources in the efficient manner.

#### 4. PROPOSED ALGORITHM

**Step 1:** Start

**Step 2:** Initialize Jobs such that  $J_i = 1$  to  $n$  and also enter the resources  $R_s$

Where  $J$  = number of tasks and  $n$  is the number of tasks limit

**Step 3:** Evaluate the time needed per job

**Step 4:** Accumulate the task Id in the array

**Step 5:** Deployment the number of users  $p = 1, 2, 3, \dots, n$  and Nodes  $N[i]$  such that  $i = 1$  to  $n$

**Step 6:** Burst time generation for the completion of the user request

**Step 7:** Array initialization  $A[x]$  where  $x = 1$  to  $N$  to store the execution of the completion of the processes

**Step 8:** IF (the process takes the time which is less than the allocated time)

The process will be completely executed in the first time

ELSE

The process will be in pending state and the whole round robin process will again execute for remaining completion of the tasks

END

**Step 9:** Evaluate the completion time of each user request and also store the ids of the completion tasks in the array

**Step 10:** Initialize population of ants as jobs and pheromones for the population

**Step 11:** For  $i = 1$ : number of iterations

For  $j = 1$ : number of ants

Evaluate Fitness and touring of the ants with fewer loads having less standard deviation. If the process having less completion time having less standard deviation will be allocated resource first and Update the pheromone

End

End

**Step 12:** Evaluate the cost, latency and minimum execution time for each ant until stopping criteria

**Step 13:** Stop

#### 5. SIMULATION RESULT

To compare the performance of the proposed approach and earlier developed approach simulation results are used. For simulation GridSim simulator is used which is a toolkit for the modelling and simulation of distributed resource management and scheduling for Grid computing [11]. In order to test the efficiency of the improved load balancing approach a comparison is done for the execution time (time utilized by PE for processing a task) of heuristic load balancing algorithm and non-heuristic load balancing algorithm.

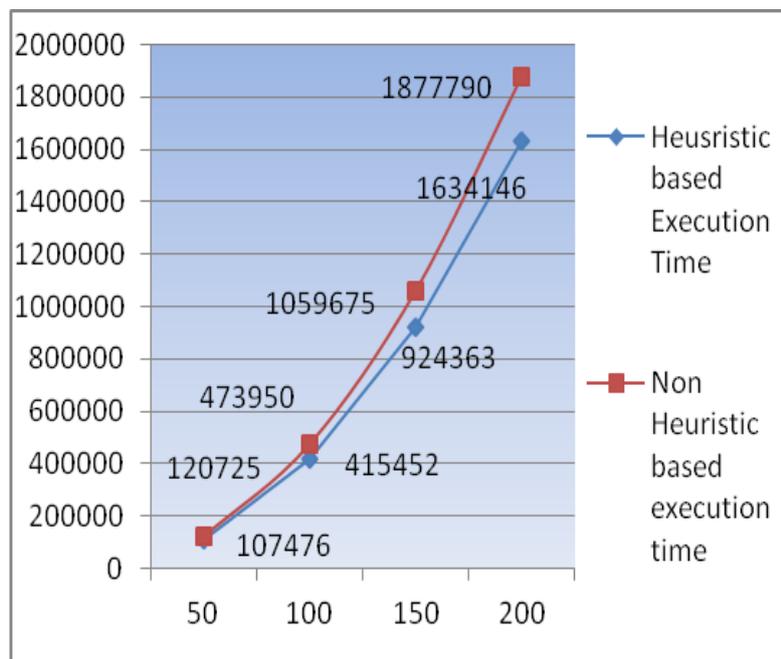
Two experiments have been conducted to evaluate the performance of the proposed algorithm in terms of their processing time. First experiment processed 50-200 jobs with fixed number of resources taken as 10 while the second experiment processed 10 jobs with varying number of resources from 5-20. Details of the scheduling parameters are shown in the Table 1.

**Table1:** Scheduling Parameters for the Experiments

Experiment	1st	2 <sup>nd</sup>
No. of machines per resource	1	1
No. of Processing Elements(PEs) per machine	1-5	1-5
MIPS rating for PEs	10 or 50 MIPS	10 or 50 MIPS
Bandwidth	1000 or 5000	1000 Or 5000

Average computation time (in seconds) has been used to compare the performance of algorithms. Fig. 3 depicts the comparison between the average computation times of the algorithms for the first experiment. In this number of resources remain fixed as 10 and the number of tasks varied from 50 to 200. It can be seen that the proposed algorithm labeled as Heuristic based load balancing algorithm has the lower execution time as compared to Non-Heuristic based load balancing algorithm at different instants.

A comparison between the computation times for the second experiment is shown in Fig. 4. Heuristic load balancing algorithm still performs better than non-heuristic load balancing algorithm. The performance of Heuristic approach is better because the communication that occur when each ant taking a step for searching the best resource is less when processing a large amount of jobs. Each ant will carry the history of visited node and will refer to it to find the best resource to process the job.



**Fig. 3.** Comparison Among the Execution Times of Heuristic and Non-Heuristic Algorithms.

In the second experiment the number of task remain fixed as 10 and number of resources varied from 5 to 20 and again compute the total execution time(in seconds) for different number of resources. Result supports the use of HRTSA for time efficient execution of tasks then NRTSA.

Fig. 3 and Fig. 4 depict the result of the total execution time for two different scenarios. Results support the proposed approach by reducing the total execution time of jobs.

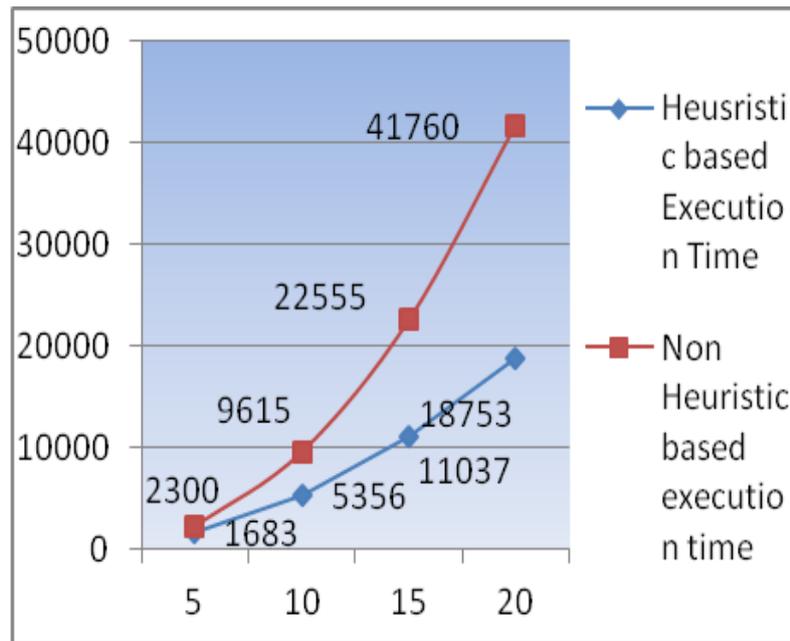


Fig. 4. Comparison Among the Execution times of Heuristic and Non-heuristic Resource Allocation Algorithms for Experiment 2.

## 6. CONCLUSION

In this research, the work is done for the resource allocation using hybrid approach of scheduling and optimization using round robin and ant colony optimization. The results shows that our proposed approach is able to achieve less execution time and also the less resource costing in grid computing systems and also our proposed approach is able to detect the variation in terms of the completion times and execution time among the users to the centralized system in an effectual manner.

## REFERENCES

- [1] A. Beloglazov, J. Abawajy, R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing", *Future generation computer systems* 28, no. 5 (2012): 755-768.
- [2] X. Fan, H. Song, J. Yang. "Imperfect information dynamic stackelberg game based resource allocation using hidden Markov for cloud computing." *IEEE Transactions on Services Computing* (2016)
- [3] A. Hameed, A. Khoshkbarforoushha, R. Ranjan, P. P. Jayaraman, J. Kolodziej, P. Balaji, S. Zeadally et al. "A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems." *Computing* 98, no. 7 (2016): 751-774
- [4] V. Talwar, S. Basu, R. Kumar, "Resource evaluation for a batch job and an interactive session concurrently executed in a grid computing environment", U.S. Patent, issued August 10, 2010.
- [5] M. Singh, Manpreet, "GRAAA: Grid Resource allocation based on ant algorithm", *Journal of advances in information technology* 1, no. 3 (2010): 133-135.
- [6] H.B. Prajapati, B. Harshadkuma., and V.A. Shah, "Scheduling in grid computing environment", In *Advanced Computing & Communication Technologies (ACCT)*, 2014 Fourth International Conference on, pp. 315-324. IEEE, 2014
- [7] M. Memoria, Minakshi, and M. Yadav, "Fault Tolerance in Grid Computing with Improved Resource Utilization and Maximum Efficiency", *International Journal of Engineering Science*, Volume 6, no.3,pp:2308-23010, 2016.
- [8] K. B. Morey, S. B. Jadhav, "A new approach for dynamic load balancing using simulation in grid computing", volume: 3, Issue: 1, pp: 256-258, 2016.

- [9] R. Nawaz, W.Y. Zhou, M. U. Shahid, and O. Khalid, "A qualitative comparison of popular middleware distributions used in grid computing environment", In Computer and Communication Systems (ICCCS), 2017 2nd International Conference on, pp. 36-40, IEEE, 2017
- [10] A. M. Vulcan, A. Mihai, M. Nicolae, "High Performance Computing Based on a Smart Grid Approach", In Control Systems and Computer Science (CSCS), 2017 21st International Conference on, pp. 651-655. IEEE, 2017
- [11] M.R. Islam, M. Rashedul, and M. N. Akhtar, "Fuzzy logic based task allocation in ant colonies under grid computing", In Electrical, Computer and Communication Engineering (ECCE), International Conference on, pp. 22-27. IEEE, 2017
- [12] Aggarwal, Er Anuj, Rajesh Verma, and Ajit Singh. "Resource Allocation Algorithm in Distributing System Using Ant Colony Optimization-A Review", International Journal of Emerging Technologies in Engineering Research (IJETER), Volume no. 4, no. 8, pp: 29-33, 2016.
- [13] J. Abawajy, and R. Buyya. "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing." Future generation computer systems 28, no. 5, pp:755-768, 2012.
- [14] Kosta, Sokol, A. Aucinas, P. Hui, R. Mortier, and X. Zhang. "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading", in Infocom, pp. 945-953, 2012
- [15] Lee, Y. Choon, A. Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems." The Journal of Supercomputing 60, no. 2, pp: 268-280, 2012.
- [16] R. Stadler, H. Lindgren, "Dynamic resource allocation with management objectives—Implementation for an OpenStack cloud", Network and service management (cnsm), pp. 309-315, IEEE, 2012.
- [17] M. Qiu, Z. Ming, G. Quan, X. Qin, "Online optimization for scheduling preemptable tasks on IaaS cloud systems", Journal of Parallel and Distributed Computing 72, no. 5, pp: 666-677, 2012.
- [18] Y. Laili, F. Tao, L. Zhang, B.R. Sarker, "A study of optimal allocation of computing resources in cloud manufacturing systems", The International Journal of Advanced Manufacturing Technology 63, no. 5, pp: 671-690, 2012.
- [19] L.F. Bittencourt, E. R.M. Madeira, and Nelson LS Da Fonseca. "Scheduling in hybrid clouds." IEEE Communications Magazine 50, no. 9 (2012)
- [20] Q. Zhang, Q. Zhu, R. Boutaba, "Dynamic resource allocation for spot markets in cloud computing environments", In Utility and Cloud Computing (UCC), Fourth IEEE International Conference on, pp. 178-185. IEEE, 2011
- [21] A. Beloglazov, J. Abawajy, R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing", Future generation computer systems 28, no. 5, pp:755-768, 2012.
- [22] A. Iosup, D. Epema, "Grid computing workloads", IEEE Internet Computing 15, no. 2, pp: 19-26, 2011.
- [23] A. Iosup, S. Ostermann, M. N. Yigitbasi, R. Prodan, Thomas Fahringer, and Dick Epema, "Performance analysis of cloud computing services for many-tasks scientific computing", IEEE Transactions on Parallel and Distributed systems 22, no. 6, pp: 931-945, 2011.
- [24] T. Kokilavani, D.I. George Amalarethnam, "Load balanced min-min algorithm for static meta-task scheduling in grid computing", International Journal of Computer Applications 20, no. 2, pp: 43-49, 2011.