Adaptive and Dynamic Load Balancing in Grid Using Ant Colony Optimization

Sandip Kumar Goyal [1], Manpreet Singh [2]
[1]Department of Computer Engineering, M.M. University,
Mullana, Ambala, Haryana (133207), India
skgmmec@gmail.com
[2]Department of Computer Engineering, M.M. University,
Mullana, Ambala, Haryana (133207), India
drmanpreetsinghin@gmail.com

Abstract— **Grid Computing involves coupled and coordinated use of geographically distributed resources for purposes such as large-scale computation and distributed data analysis. With the Grid becoming a viable high-performance alternative to the traditional supercomputing environment, a suitable and efficient load balancing algorithm is needed to equally spread the load on each computing node in the grid. This research work presents Ant based algorithm to solve the load balancing problem in computational grid. In proposed algorithm, the pheromone is associated with resources, rather than path. The increase or decrease of pheromone represent load and depends on task status at resources. The main objective of proposed algorithm is to map tasks to resources in a way that balance out the load resulting in improved utilization of resources.**

Keyword- **Grid Computing, Load Balancing, Ant Colony Optimization, Resource Utilization.**

I. Introduction

Grid infrastructure integrates large computational and storage resources, data, services and applications from different disciplines [1], [2]. Due to random task arrival patterns and heterogeneous nature of resources, resources in one grid site may be over-loaded while others in a different grid site may be under-loaded. It is therefore, required to dispatch tasks to idle or under-loaded sites to obtain better resource utilization and reduce the average task response time. Task scheduling [3] and load balancing [4] are key grid services, where issues of load balancing represent a common concern for most grid infrastructure developers.

Load-balancing algorithms can be classified into static and dynamic approaches. Static load-balancing algorithms [5], [24] assumes that a priori information about all the characteristics of the tasks, the computing nodes and the communication network are known and provided. In contrast, dynamic load-balancing algorithms [6], [24] attempt to use the runtime state information to make more informative decisions in sharing the system load. It is now commonly agreed that despite the higher runtime complexity, dynamic algorithms can potentially provide better performance than static algorithms.

Ant Colony Optimization (ACO) is a relatively new computational and behavioural paradigm for solving optimization and combinatory problems; it is based on the principles that control the behaviour of natural systems. In this paper, Ant based algorithm for load balancing in Grid is proposed. The research work aims on improving the way ants search the best resources in terms of minimizing the processing time of each task and at the same time balancing the load on available resources.

II. Ant Colony Optimization

Many aspects of the collective activities of social insects, such as ants, are self-organizing. This means that complex group behaviour emerges from the interactions of individuals who exhibit simple behaviours by themselves. Examples of these collective activities among ants are finding food and building nests [7]. The results of self-organization are global in nature, but come about from interactions based entirely on local information. To achieve this, self-organization relies on several components: (i) positive feedback (ii) negative feedback (iii) multiple interactions.

The capabilities of a single ant are very limited compared to those of a colony. In some species, ants are mostly blind and they communicate poorly. But collectively, ants can establish the shortest route between a source of food and their nest and efficiently move the food to their home [9]. Ants communicate with each other through the use of pheromones. As ants traverse a path, they deposit pheromones. Pheromones are chemical substances that attract other ants and are deposited by ants on the ground as they travel. Ants move randomly, but when they encounter a pheromone trail, they decide whether or not to follow it. If they do so, they lay down their own pheromone on the trail as well, reinforcing the pathway. The probability that an ant chooses one path over other increases proportional to the amount of pheromone present. The more ants that use a given trail, the more attractive that trail becomes to subsequent ants.

Ant Colony Optimization (ACO) is a meta-heuristic using artificial ant to find desirable solutions to difficult combinatorial optimization problems [7]. The behaviour of artificial ants is based on the traits of real ants as described above, plus additional capabilities that make them more effective, such as a memory of past actions. Each ant of the "colony" builds a solution to the problem under consideration and uses information collected on the problem characteristics and its own performance to change how other ants see the problem [8].

III. RELATED WORK

In the past decade, a lot of research has been directed towards the task of effective load balancing algorithms [10] for distributed computing systems. Generally, there are two methods for creating load balancing in Grid environment: static load balancing and dynamic load balancing. In some algorithms the combination of these two methods are used which are referred as combined algorithms.

One of these combined algorithms is the load balancing algorithm based on the table of effective nodes [11]. Another combined algorithm proposed for load balancing is Basic Hybrid algorithm [12] which used the combination of two methods Deferred and Random. The Deferred method uses the dynamic information of the site and the Random method uses the static information of the site. Some of the algorithms have been developed by using genetic approach through which the selection of the nodes are done by genetic operators which include three operators of reproduction, exchange and mutation [16]. There are also a number of algorithms that use the tree method for load balancing [17].

A recent approach is the use of ACO for scheduling jobs in grid [13]. ACO algorithm is used in grid computing because it is easily adapted to solve both static and dynamic combinatorial optimization problems. In [14], ACO has been used as an effective algorithm in solving the load balancing problem in grid computing. A study in [15] proposed a new algorithm that is based on an echo intelligent system, autonomous and cooperative ants. In this algorithm, the ants can procreate and also can commit suicide depending on existing condition. Ant level load balancing is proposed to improve the performance of the mechanism. ACO algorithm for load balancing in distributed systems through the use of multiple ant colonies is proposed in [20]. In this algorithm, information on resources is dynamically updated at each ant movement. The study to improve ant algorithm for job scheduling in grid computing is based on the basic idea of ACO was proposed in [21]. The pheromone update function in this research is performed by adding encouragement, punishment coefficient and load balancing factor. Balanced job assignment based on ant algorithm for computing grids called BACO was proposed in [22].

A game-theoretic-based solution [18] to the grid load-balancing problem is proposed. The developed algorithm combines the inherent efficiency of the centralized approach and the fault-tolerant nature of the decentralized approach. [17] proposed a framework consisting of distributed dynamic load balancing algorithm in perspective to minimize the average response time of applications submitted to grid computing. In [7], authors has presented a Multiple Ant Colony Optimization (MACO) approach for load balancing in circuit–switched networks. MACO uses multiple ant colonies to search for alternatives to an optimal path. Each group of mobile agents corresponds to a colony of ants, and the routing table of each group corresponds to a pheromone table of each colony. Route [23] is a load balancing algorithm addressed to grid computing environments where there is a large amount of resources, heterogeneity, high communication latency, large number of users and distributed location.

An enhanced ant algorithm for load balancing in grid computing is proposed in [9]. The proposed algorithm will determine the best resource to be allocated to the jobs based on job characteristics and resource capacity, and at the same time to balance the entire resources. In [19], dynamic grid scheduling algorithm based on adaptive ant colony algorithm was proposed. In this algorithm, the evaporation rate value was adaptively changed and a minimum value of zero was fixed. The local and global pheromone updates were used in order to control the pheromone value of each resource.

IV. ANT BASED LOAD BALANCING ALGORITHM (ALBA)

In proposed ant algorithm, the pheromone is associated with resources, rather than path. The increase or decrease of pheromone represent load and depends on task status at resources. The notations used in the description of ALBA are illustrated in Table I.

TABLE I Notations Used

T	Number of Tasks
R_i	i^{th} Available Resource
$\tau_i(0)$	Initial Pheromone Associated with R_i
$\tau_i(t)$	Current Pheromone Associated with Ri
$p_i(t)$	Probability of Task Assignment to Ri
α	Relative Performance of Pheromone Trail Intensity
β	Relative Importance of Initial Performance Attributes
ρ	Pheromone Decay Parameter with Value Between 0 and 1
Δ_i	Pheromone Variance
N	Number of Resources
FT_i^t	Finish Time of Task t on Resource Ri
RU_i	Resource Utilization of Ri

The working of proposed algorithm is as follow:

Step 1. Initialize the value of α, β, ρ, Δ, N, T, RU_i and also set pheromone trails for each resource.

Step 2. Select the next task *t*.

Step 3. Determine the transition probability (load) of each resource R_j as:

$$p_j(t) = \frac{[\tau_j(t)]^{\alpha} * [\eta_j]^{\beta}}{\sum_r [\tau_r(t)]^{\alpha} * [\eta_r]^{\beta}}$$

Step 4. Find resource R_i with high transition probability among all resources:

$$p_i(t) = \max_{l \in N} p_l(t)$$

i.e. resource R_i is having minimum load.

Step 5. Assign task *t* to R_i.

Step 6. Set T = T - 1.

Step 7. Check whether any task completion or failure reported. If no, go to Step 11.

Step 8. If (task completion at any resource R_i) then Increase pheromone of R_i as:

$\tau_i(t) = \tau_i(t) + \Delta$ reporting R_i as lightly loaded.

Step 9. $RU_i = RU_i + FT_i^t$

Step 10. If (task failure at any resource R_i) then decrease pheromone of R_i as:

$$\tau_i(t) = \tau_i(t) - \Delta$$

reporting R_i as heavily loaded.

Step 11. If (T>0) then go to Step 2.

Step 12. For each resource R_i, $1 \le i \le N$

Compute $RU_i = \dfrac{RU_i}{\sum_{k=1}^{N} RU_k}$

Print resource utilization of R_i.

V. SIMULATION RESULTS & DISCUSSION

In this section, some experiments that have been carried out to test the efficiency and effectiveness of proposed algorithm are presented. The functional code is implemented using GridSim on an Intel core 2 duo, 2 GHz window based laptop. Table II specifies the simulation environment:

TABLE II Simulation Parameters

Simulation Runs	10
No. of Resources	10 – 30
No. of Tasks	25 – 100
Task Size (MI)	10000 - 100000
Processing Power of Resources(MIPS)	300 - 1000

In order to determine whether ALBA can search a near optimal schedule for a large number of tasks or resources, the simulation was performed in two scenarios.

A. Scenario 1 (Effect of load in terms of tasks on average resource utilization)

The number of tasks is varied from 25 to 100 while keeping resources as 10 and the result on resource utilization is depicted in Fig.1, 3, 4 and 5. The individual assignment of 25 tasks to 10 resources under proposed algorithm and WALBA (Without Ant based Load Balancing Algorithm) is shown in Fig. 2.

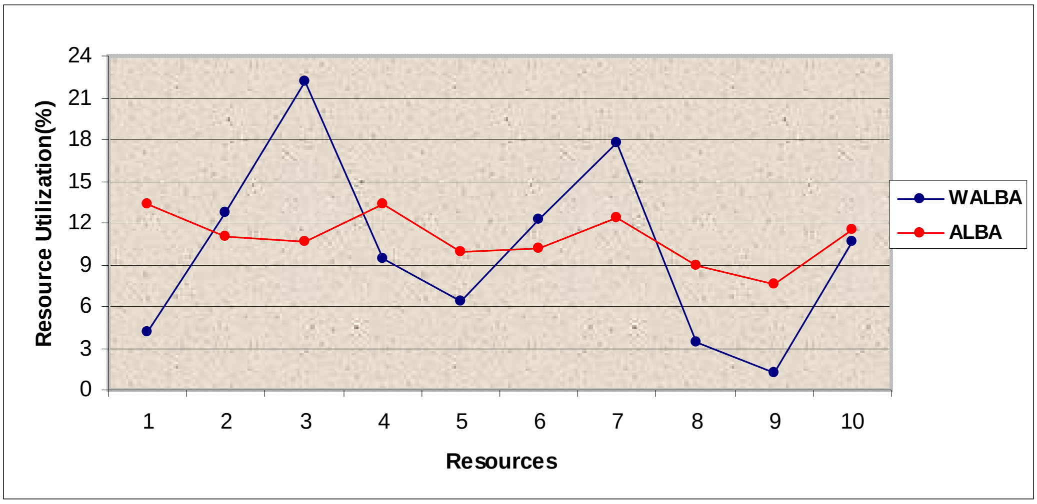


Fig. 1. Effect of load variation on average resource utilization with Tasks = 25 and Resources = 10.

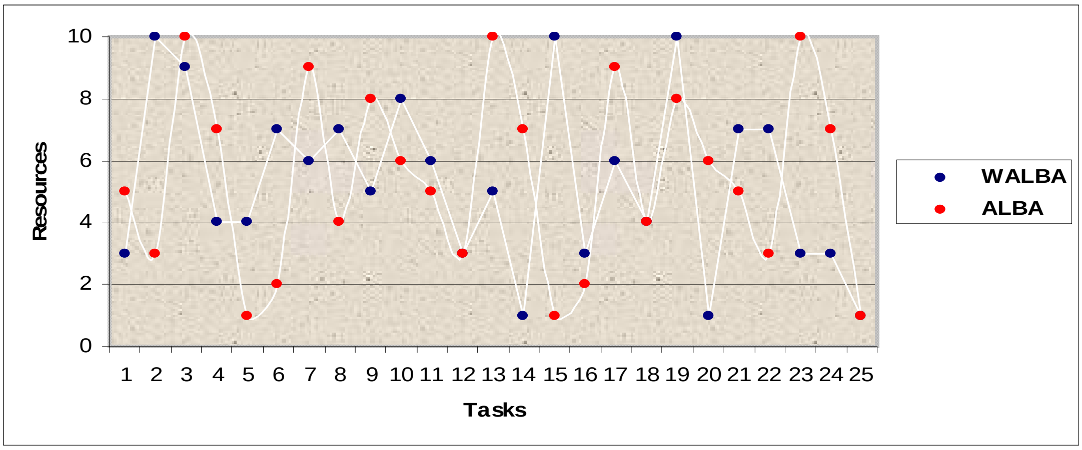


Fig. 2. Assignment of Tasks with Tasks = 25 and Resources = 10.

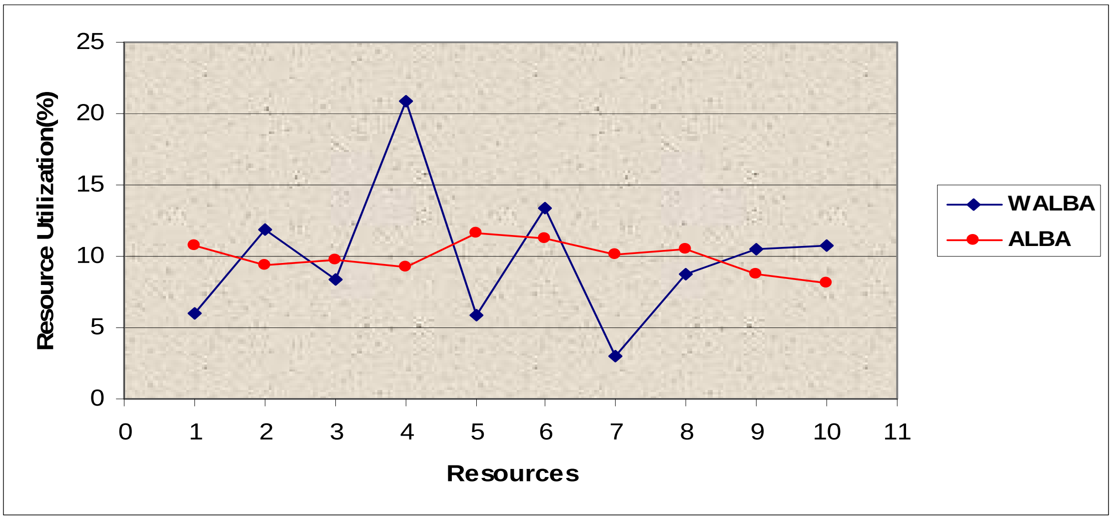


Fig. 3. Effect of load variation on average resource utilization with Tasks = 50 and Resources = 10.

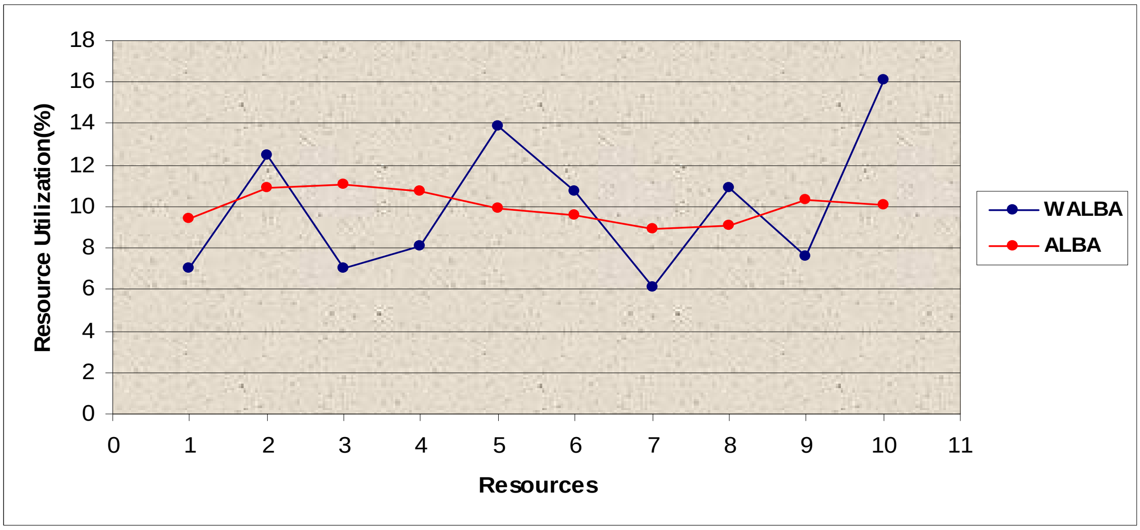


Fig. 4. Effect of load variation on average resource utilization with Tasks = 75 and Resources = 10.

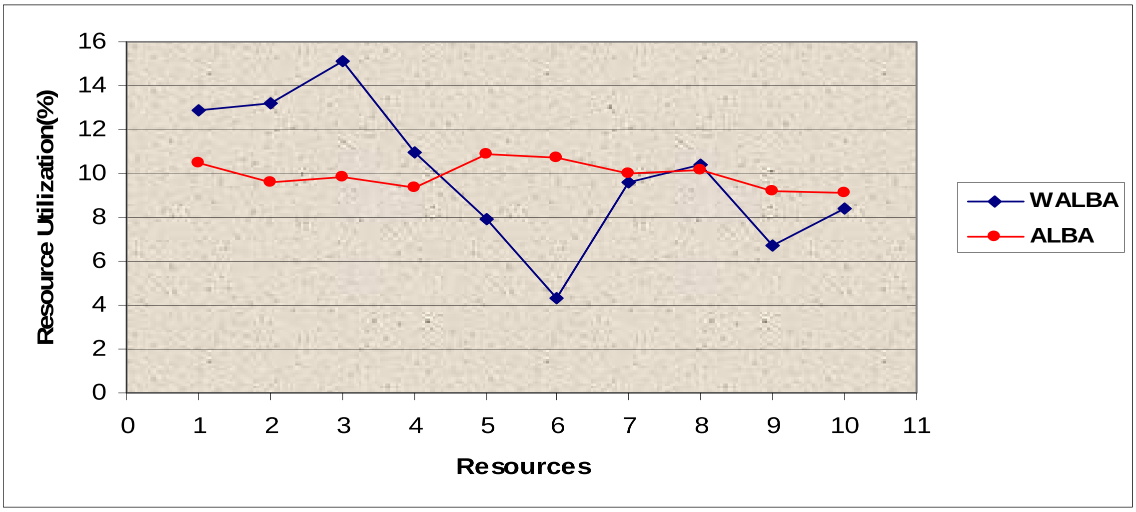


Fig. 5. Effect of load variation on average resource utilization with Tasks = 100 and Resources = 10.

This improvement is expected as ALBA is keeping track of the state of all resources at each point in time which makes it able to take optimal decisions.

B. Scenario 2 (Effect of scalability on average resource utilization)

The number of resources is varied from 10 to 30 while keeping number of tasks as 75. Fig. 6 and 7 shows the effect of scalability on the performance of algorithms under observation in terms of load balancing.

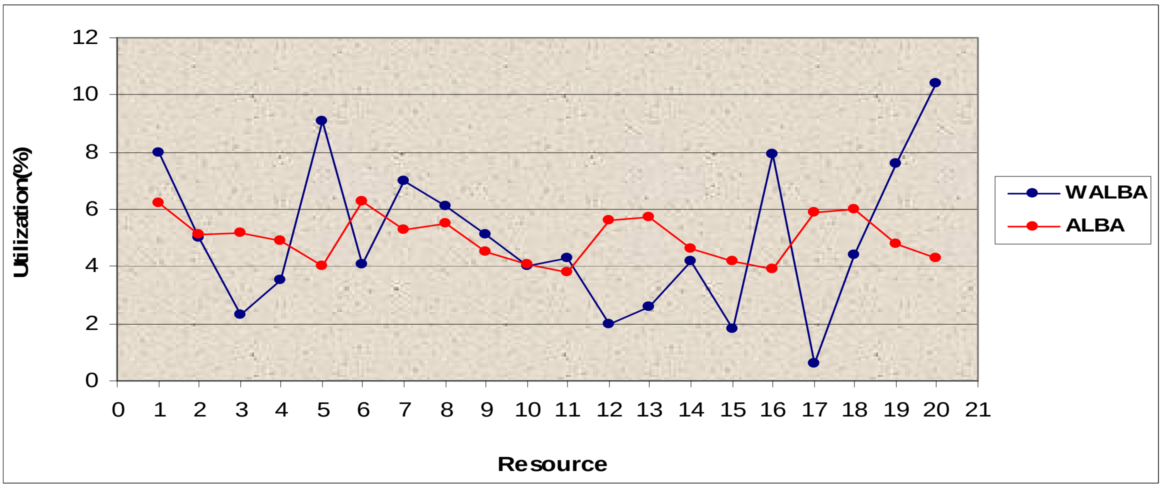


Fig. 6. Effect of scalability with Tasks = 75 and Resources = 20.

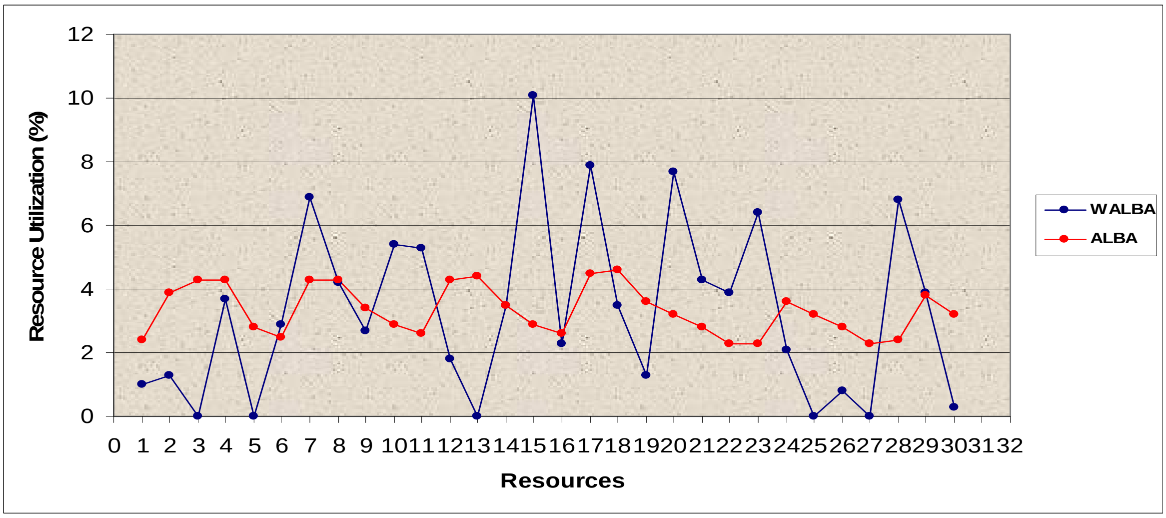


Fig. 7. Effect of scalability with Tasks = 75 and Resources = 30.

Fig. 8 shows that proposed algorithm is better than WALBA as the standard deviation for ALBA is not more than 0.77 and the standard deviation for WALBA ranges from 2.35 to 3.18.

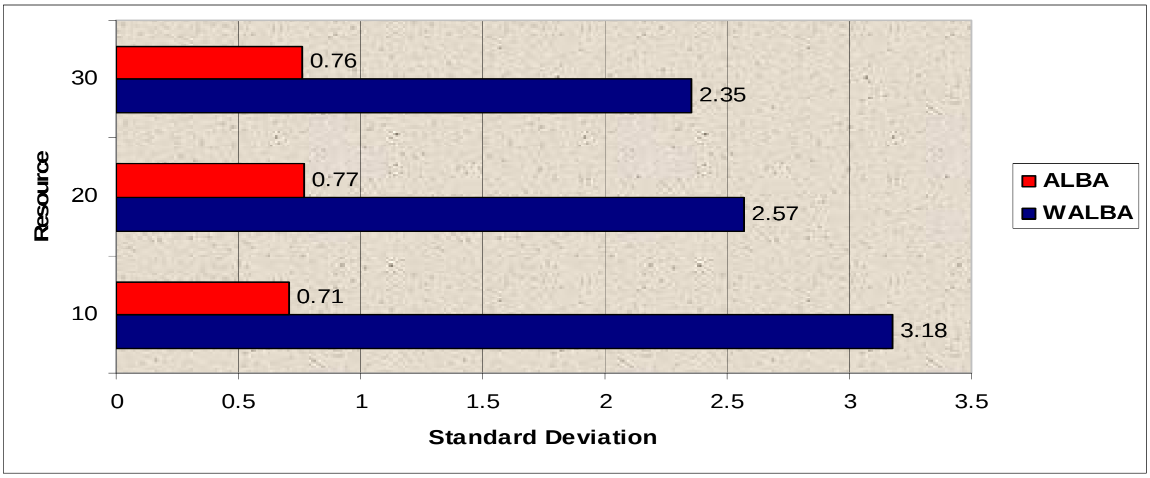


Fig. 8. Comparison in terms of standard deviation with Tasks = 75.

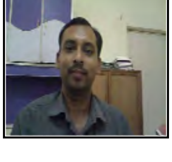
VI. Conclusion

Load balancing is one of the main issues in the grid environment. Recent researches have proved that load balancing on computational grids is best solved by heuristic approach. Hence, an ant based load balancing algorithm is developed to allocate tasks to proper resources. In order to verify the performance of proposed algorithm, the simulation is performed. The results of the experiments are also presented and the strength of the algorithm is investigated. The simulation result concludes that the proposed algorithm enhances performance in terms of resource utilization.

REFERENCES

[1] I. Foster, and C. Kesselman, *The Grid: Blueprint for a new Computing Infrastructure*, 2nd ed.: Morgan Kauffman publishers, 2004.
[2] M. Bote-Lorenzo, Y. Dimitriadis, and E. Gomez-Sanchez, "Grid characteristics and uses: a grid definition," in *Proc. 1st European Across Grids Conference (ACG'03)*, 2004, pp. 291-298.
[3] A.Y.Zomaya, and Y.H.The, "Observations on using genetic algorithms for dynamic load-balancing," *IEEE Transactions on Parallel and Distributed Systems*, Vol.12, No. 9, pp. 899-912, 2001.
[4] M. Singh, and P.K. Suri, "An efficient decentralized load balancing algorithm for grid," in *Proc. 2nd IEEE Inter. Conf. (IACC)*, 2010, pp. 10-13.
[5] A.N.Tantawi, and D. Towsley, " Optimal static load balancing in distributed computer systems," *Journal of Association for Computing Machinery*, Vol. 32, No. 2, pp. 445-465, April 1985.
[6] J. Xu, and K. Hwang, "Heuristic methods for dynamic load balancing in a message-passing multicomputer," *Journal of Parallel and Distributed Computing*, Vol. 18, pp. 1-13, 1993.
[7] A. D. Ali, and M. A. Belal, "Multiple ant colonies optimization for load balancing in distributed systems," in *Proc. Inter. Conf. (ICTA'07)*, 2007.
[8] P. McMullen and P. Tarasewich, "Using ant techniques to solve the assembly line balancing problem," Institute of Industrial Engineers *Trans.*, vol. 35, no. 7, pp. 605-617, 2003.
[9] H. J. A. Nasir, K. R. K. Mahamud, and A. M. Din, "Load balancing using enhanced ant algorithm in grid computing," in *Proc. 2nd Inter. Conf. Computational Intelligence, Modelling and Simulation*, 2010, pp. 160-165.
[10] L. M. Khanli, and B. Didevar, "A new hybrid load balancing algorithm in grid computing systems," *International Journal Computer Science and Technology*, vol. 2, no. 5, 2011.
[11] K. Q. Yan, S. C. Wang, C.P Chang, and J. S. Lin, "A hybrid load balancing policy underlying grid computing environment," *Computer Standards and Interfaces*, vol. 29, no. 2, pp. 161-173, 2007.
[12] S. Zikos, and H. D. Karatza, "Communication cost effective scheduling policies of nonclairvoyant jobs with load balancing in a grid," *Journal of Systems and Software*, vol. 82, no. 12, pp. 2103-2116, 2009.
[13] S. Fidanova, and M. Durchova, "Ant algorithm for grid scheduling problem," in *Proc. 5th Inter. Conf. Large Scale Scientific Computing (LSSC)*, vol. 3743, 2006, pp. 405-412.
[14] Y. Li, "A Bio-inspired adaptive job scheduling mechanism on a computational grid," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 6, no. 3, pp. 1-7, 2006.
[15] M. Salehi, and H. Deldari, "Grid load balancing using an echo system of intelligent ants," in *Proc. 24th Inter. Conf. Parallel and Distributed Computing and Networks*, 2006, pp. 47-52.
[16] Y. Li, Y. Yang, M. Ma, and L. Zhou, "A hybrid load balancing strategy of sequential tasks for grid computing environments," *Future Generation Computer Systems*, vol. 25, no. 8, pp. 819-828, 2009.
[17] B. Yagoubi, and Y. Slimani, "Task load balancing strategy for grid computing," *Journal of Computer Science*, vol. 3, no. 3, pp. 186-194, 2007.
[18] R. Subrata, A. Y. Zomaya, and B. Landfeldt, "Game theoretic approach for load balancing in computational grids," *IEEE T.rans. Parallel and Distributed Systems*, vol. 19, no. 1, pp. 66-76, 2008.
[19] Liu., A. Wang, " Grid task scheduling based on adaptive ant colony algorithm," in *Proc. Inter. Conf. Management of e-commerce & e-Government*, 2008, pp. 415-418.
[20] A. Ali, M. A. Belal, and M. B. Al-Zoubi, "Load balancing of distributed system based on multiple ant colonies optimization," *Journal of Applied Sciences*, vol. 7, no. 3, pp. 433-438, 2010.
[21] H. Yan, X. Shen, X. Li, and M. Wu, "An improved ant algorithm for job scheduling in grid computing," in *Proc. 4th Inter. Conf. Machine Learning and Cybernetics*, 2005, pp. 2957-2961.
[22] R. Chang, J. Chang, and P. Lin, "Balanced job assignment based on ant algorithm for grid computing," in *Proc 2nd IEEE Asia-Pacific Service Computing Conference (APSCC)*, 2007, pp. 291-295.
[23] R. F. Mello, L. J. Senger, and L. T. Yang, "A routing load balancing policy for grid computing environments," in *Proc. 20th Inter. Conf. Advanced Information Networking and Applications (AINA'06)*, 2006, pp. 153-158.
[24] S. K. Goyal, R. B. Patel, and M. Singh, "Adaptive and dynamic load balancing methodologies for distributed environment: a review," *International Journal of Engineering Science and Technology (IJEST)*, vol. 3, no. 3, pp. 1835-1840, 2011.

AUTHOR'S BIOGRAPHY



Sandip Kumar Goyal received his B.Tech, M.Tech. from Kurukshetra University, Kurukshetra, India and is currently enrolled as a Ph.D. scholar in the department of Computer Science and Engineering at M.M.University, Mullana, Ambala, Haryana, India. He is presently serving as Assoc. Professor in Computer Engineering Department of M.M. Engineering College, Mullana, Ambala. He is in teaching since 2000. He has published 10 research papers in International and National journals and conferences. His research area is Load balancing Methodologies in distributed environment.

Dr. Manpreet Singh received his B. Tech., M.Tech. and Ph.D. from Kurukshetra University, Kurukshetra, India. He is presently serving as Professor and Head, Computer Science and Engineering Department of M. M. Engineering College, Mullana, Ambala. He has about 13 years of experience in teaching and research. He has published 30 research papers in International and National Journals and Conferences. His is current research interest includes Grid Computing, Cloud Computing and MANETs.