Journal of Theoretical and Applied Information Technology
20th September 2014. Vol. 67 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IMPROVED GENETIC ALGORITHM FOR GROUP-BASED JOB SCHEDULING IN MOBILE GRIDS

[1]G.SARAVANAN, [2]Dr.V.GOPALAKRISHANAN
[1]Assistant Professor, Computer Science and Engineering
Excel Engineering College, Salem Main Road, NH 47,
Pallakapalayam, Komarapalayam – 637 303
[2]Associate Professor, Department of Electrical and Electronics Engineering
Government College of Technology,
Thadagam Road, Coimbatore
E-mail: [1]saravanang0681@gmail.com

ABSTRACT

In mobile grids, the existing job scheduling scheme causes increased job processing time. The increased processing time and overhead may results in system degradation. Also mobility and resource availability parameters are not considered during job scheduling. Hence in this paper, we propose an improved genetic algorithm for group-based job scheduling in mobile grids. In this technique, the jobs are grouped according to the resource availability. Then the jobs are scheduled based on the parameters such as mobility, resource availability and job completion time using enhanced genetic algorithm. By simulation results, we show that the proposed technique minimizes the job completion time, thus enhancing the system performance and minimizing the overhead time.

Keywords: *Genetic Algorithm, Job Scheduling, Mobile Grids, Simulation, Overhead Time*

1. INTRODUCTION

1.1 Mobile Grids

Grid is already being successfully used in many scientific applications where huge amounts of data have to be processed and/or stored. Mobile Grid is a platform that addresses mobility issues by means of enabling both fixed and mobile users to have access to both fixed and mobile Grid resources by utilising the underlying technologies transparently and efficiently. Mobility involves a set of issues for both users and communication/collaboration mode that will be presented shortly in the sequel [1].

A mobile device in mobile Grid can play roles as both a resource consumer and a service provider. As a consumer it requests service to a Grid, and as a provider it actively participates in processing service requests. Compared to physically fixed Grid resources such as desktop computers, mobile devices tend to provide a relatively inferior performance in terms of CPU capability, amount of main memory, and storage capacity. They also have a limited battery life and are commonly connected to wireless networks that are not as reliable as wired networks.

Considering the devices capabilities and enormous number of gadgets, mobile devices have immense potential to serve as resource providers in mobile Grid environment. The challenges in mobile Grid are job scheduling and load balancing issues under the unreliable communication environment [2].

1.2 Scheduling in mobile grids

The main goal of grid is providing services with high reliability and lowest cost for large volumes of users and support group work and the most important issue in grid computing are resource management and control, reliability and security. It is one of the important issues to increase the efficiency of the grid. To increase the efficiency of grid a properly and useful scheduling is needed. Unfortunately, the dynamic nature of grid resources and the demands of different users are causing complexity in the scheduling grid. Scheduling process in grid can be organized in three stages: resource discovery, resource selection and scheduling based on clear goals and the last stage is request assignment to the most appropriate source [3].

Journal of Theoretical and Applied Information Technology
20th September 2014. Vol. 67 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

1.3 Job scheduling in mobile grids

The scheduler should allocate jobs to proper mobile devices, minimize uncertainty in job execution and strive to optimize scheduling objectives such as maximize throughput, response time and balance available resources. Since the mobile grid is much less stable than the wired environment, dynamism should be considered at the scheduling time. Job scheduling in mobile grids thus require a robust system model that can incorporate all these factors. In order to meet the dynamic and mobile nature of resources, the availability should be predicted incase of scheduling [4].

1.4 Issues in scheduling

Load balancing policies subject to the underlying mobility were based on a small part of the offered resources is wasted and a small part of the workload has to be processed again. The size of the installments increases as the size of the aborted fragments of the workload increases. A dynamic load balancing strategies increases the communication overhead due to exchanging information among nodes [12] [13].

Resource discovery activity involves searching resources from resource providers. The resource for the appropriate resource types that match the user's providers are represented by Resource-owner Agent application requirements. But sometimes it is not able to provide an efficient resource discovery. Fault tolerance is sometimes not realiable which in turn leads to run time overhead.

Also in job scheduling there is degradation in application and communication performance due to the unstable network connection. Other than this some of the issues are as follows

- Heterogeneity
- Utility
- Scalability
- Limited energy
- Resource reliability [11] [14]

1.5 Problem Identification

In [15] a modified grouping-based job scheduling in computational grid is proposed. The main objective of this proposed solution is to minimize the overhead time and computation time which also reduces the overall processing time of jobs. The motivation of this paper is to develop an enhanced grouping based job scheduler and grid resource allocation algorithm that must be efficient and effective in reducing the total dealing out time of the jobs.

The main process of the scheduler is to achieve the resources with the goal of maximizing throughput, efficiency, resource utilization, minimizing job completion time, communication overhead and cost or both time-cost etc., Most Scheduling algorithms interact with grid information service (GIS) to obtain the initial list of authorized resources, called resource pool. The scheduler is used to select those resources that are expected to meet time, cost or both time-cost and any other additional constraints enforced by the user. The resource selection along with job-grouping by the scheduler is based on some efficient and low cost strategy. The grid resource algorithm used in this paper reduces the processing time of all jobs.

In scheduling the jobs, job completion time, mobility and availability parameters are not considered in this scheme. The simulation results shows that the MGBJS gives better performance compared to other scheduling processes in processing and overhead time. However the overall performance in terms of overhead is only upto 8.4 % and processing is 18.3%. Hence there is a slight degradation in the performance of the system.

In [10], a task scheduling method based on genetic algorithm (GA) is proposed. This in turn shortens the time span and improves the utilization ratio of resources. The genetic algorithm is introduced to consider the mobile grid task scheduling. Here the scheduling method which is based on Genetic Algorithm is well-known and robust search technique for larger scale optimization problems. The fitness function measures the quality of each solution and assigns the possibility of inheritance to the next generation for each individual. The time fitness function estimates the completion time of each individual. However the fitness function is not computed by means of the parameters such as availability and mobility.

As a solution to the above problems, we propose to design enhanced genetic algorithm for grouping-based job scheduling.

2. RELATED WORK

JongHyuk Lee et al [2] presents a novel balanced scheduling algorithm in mobile Grid, taking into account the mobility and availability in scheduling. They have also proposed a load balancing technique by classifying mobile devices into nine groups depending on availability. The experimental results show that their scheduling algorithm

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

provides a superior performance in terms of execution times to one without considering availability and load-balancing. However it provides only 10% of performance improvement compared to a prior work.

S. Stephen Vaithiya et al [4] have proposed a task scheduling algorithm based on the dynamic prediction of resource mobility and battery power in the mobile grid environment. The mobility is predicted based on the movement type, movement pattern and movement direction of the mobile resource within the zone. The battery power is predicted based on the C-rate of the mobile resource. The availability prediction of a mobile resource is evaluated during the task submission and this allows the system to consider run-time parameters prior to execution. The simulation results point to the efficacy of the proposed work. However when the resource has the higher priority the resource prediction time is reduced which in turn reduces the total execution time of the task.

Liu Lei and Li Chunlin [5] have proposed the resource reliability which is evaluated from task completion time and energy consumption, in order to solve the problems caused by mobile devices in the scheduling. Moreover, the grid tasks are divided into two categories: urgent task and non-urgent task, which adopt different ways to calculate the resource reliability. Then, the revised Min-Min algorithm is proposed, which concerns with the resource reliability. It aims to select the most reliable resource to execute the grid tasks and ensure successful scheduling. Results show that it can effectively shorten the task completion time by reducing the resource energy consumption and guarenting the load balancing. However, the resource reliability is reduced with the increase of tasks received.

M. N. Birjea and S. S. Manvi [6] have proposed a scheme for seamless job scheduling by using software agents in bandwidth constrained wireless grids. An economic scheme by using non-cooperative bargaining game is designed to encourage resource sharing depending upon grid market dynamics. The scheme consists of two agencies: grid information service and resource broker agency. Based on the mobility of a device, agents decide dynamically either to continue or terminate execution of a scheduled job. The scheme is simulated to evaluate the performance parameters such as expected surplus, job completion time and job execution rate. However the communication and negotiation delays become negligible and processing delay carries more weightage with higher availability of bandwidth.

Juan Manuel Rodriguez et al [7] have proposed a novel job scheduler that aims to use the energy in an efficient way. To implement it, they developed a simple but effective battery estimation model. In some experiments their energy aware scheduler outperformed traditional Grid schedulers to effectively address energy constrains in mobile Grids. Furthermore, the simulations also put in evidence that all the evaluated scheduling algorithms are more effective. From this work it is evident that there are a range of complex issues that if properly solved would enable the development of a more effective scheduler for Grid systems that use mobile devices. One of the problems is to estimate the capacity of a mobile device. This problem is related not only with estimating the remaining battery time, which already is a complex problem but also estimating the amount of work that a mobile device can do in that time. Another problem is to estimate how complex a job is, because better estimations allow the scheduler to calculate how many resources a particular work will consume, allowing a better resource assignation.

Du Li-juan and Yu Zhen-wei [8] have proposed MGRR algorithm for scheduling independent tasks with same length onto mobile grid with dynamic performance resources. Making use of re-scheduling and replication, the proposed algorithm does not use any prediction information on the performance of resources and adapts to intermittent connectivity of mobile resource. The MGRR algorithm's objective function is total processor cycle consumption, which is the total number of instructions consumed for a application during schedule. The algorithm is well adaptive to the dynamic performance and the intermittent connection of mobile resources. Scheduling process reflects the thinking of re-scheduling and replication. Simulation result shows the influence of intermittent connection on scheduling. However as communication capability enhances, the computing power required for certain task amount reduces.

Ashish Chandak et al [9] have proposed a model which manages idle mobile nodes and fix computing nodes for task scheduling. They exploit the potential of mobile devices by taking them into consideration in grid application. In their model, mobile nodes are used with sites and they can distribute task among themselves if computation is not possible by single mobile node. They measure utility of grid system in which tasks is scheduled

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

using different heuristics. Since their model uses idle mobile nodes effectively, their simulation results show that the application utility gets increased when considering the computing capability of mobile node and site together. However there is complexity in the system.

Qing Jiang et al [10] have proposed a task scheduling method based on genetic algorithm to shorten the time span and improve the utilization ratio of resources. By means of simulation experiments show that, using genetic algorithms for mobile grid task scheduling can reduce the system total execution time and task completion time, improve grid task analysis efficiency, and make a mobile grid system scheduling performance better. However with the gradual increase in number of tasks, the overall span of lifetime will have great change.

Manoj Kumar Mishra et al [15] have proposed a modified grouping-based job scheduling in computational grid with the objective of minimizing overhead time and computation time, by reducing overall processing time of jobs. The complexity of the of the proposed dynamic job scheduling algorithm is O (nlogn) considering sorting of resources and Gridlets according to their MIPS and MI respectively, without which it will run in linear time. The overall performance improvement is up to 18.3% and 8.4% in terms of processing and overhead time respectively. However the performance of the algorithm is only from 4.2% to 8.4% in terms of overhead time.

Jong Hyuk Lee et al [16] have presented a novel balanced scheduling algorithm in mobile Grid, taking into account the mobility and availability in scheduling. They have also analyzed user's mobility patterns to quantitatively measure the resource availability that is classified into three types: full availability, partial availability, and unavailability. They also propose a load balancing technique by classifying mobile devices into nine groups depending on availability. The experimental results show that their scheduling algorithm provides a superior performance in terms of execution times to one without considering availability and load-balancing. However dynamic scheduling algorithm is not directly applicable to batch jobs. Since the performance of batch jobs would be greatly enhanced by applying minmin and max-min according to the job type.

3. PROPOSED SOLUTION

3.1 Overview

In this paper, we propose an improved genetic algorithm for group-based job scheduling in mobile grids. In this technique, the jobs are grouped according to the resource availability. Then the jobs are scheduled based on the parameters such as mobility, resource availability and job completion time using enhanced genetic algorithm. This technique minimizes the job completion time, thus enhancing the system performance.

3.2 Genetic Algorithm

Genetic algorithm is familiar and robust search technique for large scale optimization problems. It involves the operation model based on the biological evolution such as crossover, mutation and selection, striving to discover a near optimal solution.

The steps involved in genetic algorithm are as follows

Begin
Creation an initial population
Computing of fitness of each individual
While (not stopping condition) do
Select parents from population.
Execute crossover to produce offsprings.
Perform mutations.
Compute fitness of each individual.
Replace the parents by the corresponding offsprings in new generation.
End if
End if

3.3 Estimation of Metrics

Let T_u be the predicted uptime
Let T_d be the predicted downtime
Let T_c be the time during which a network is connected
Let T_{dc} be the time during which a network is disconnected.
Let T_{ij} be the time spent by i^{th} user j^{th} access point

In mobile environment, the total time of mobile device is divided into T_u and T_d. T_u is further divided into T_c and T_{dc}. Based on the defined time, the terms Resource availability and Mobility is defined as shown below.

Resource availability (RA): When the user can utilize the system instantly at a specific time is termed as availability. It is ratio of the predicted uptime to the sum of the predicted uptime and downtime. Uptime and downtime are the system power status such as ON and OFF respectively.

Journal of Theoretical and Applied Information Technology
20th September 2014. Vol. 67 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$RA_i = \frac{T_u}{T_u + T_d} \quad (1)$$

Mobility: Mobility (MO_i) is defined using two parameters such as access point prevalence (α_{ij}) and user persistence (μ_{ij}).

The access point prevalence is defined using the following Eq (2)

$$\alpha_{ij} = \frac{T_{ij}}{T_c^i} \quad (2)$$

The user persistence is defined as time duration at which the i^{th} user remains in j^{th} access point until the user moves to another access point (AP) or when the network link is down. It is shown using the following Eq (3)

$$\sum_{k=1}^{n} \mu_{ij} = T_{ij} \quad (3)$$

Thus, higher the access point prevalence and user persistence, minimum will be the mobility.

Fitness function: The fitness function of the chromosome is computed based on the mobility, resource availability and job completion time using Eq. (4).

$$F_i = \beta_1 (T_{JC}) + \beta_2 (RA) + \beta_3 (MO_i) \quad (4)$$

where β_1, β_2 and β_3 be the transformation probability subjected to Poisson distribution.

3.4 Genetic Algorithm for Group-Based Job Scheduling

Our proposed technique involves two phases
Phase I: Grouping of Jobs
Phase II: Genetic Algorithm based Job Scheduling

3.4.1 Grouping of Jobs

Let GJ be the grouped job
Let γ be the resource's processing capability measured in MIPS Million Instruction per Second
Let P be the computational power required by the job measured using Million Instructions
Let t be the user defined time used to measure total amount of GJ completed within a specific time.
Let MJ be the memory size of GJ
Let M be the memory available at resources
Let BW be the bandwidth capacity of the resources
Let t_c be the communication time of the time
Let T_{GJx} be the processing time of x^{th} GJ
Let T_{ox} be the overhead time of x^{th} GJ
Let T_{cx} be the computation time of x^{th} GJ

The grouping of jobs depends on the resources selection and job grouping technique and need to satisfy the following condition.

$$P \leq \gamma * t \quad (5)$$

$$MJ \leq M \quad (6)$$

$$MJ \leq BW * t_c \quad (7)$$

Eq. (1) reveals that P should not exceed γ.
Eq. (2) reveals MJ should not exceed M.
Eq. (3) reveals that MJ should not exceed to BW within specific time period.

The job completion time (T_{JC}) is estimated using the following Eq (8)

$$T_{JC} = \sum_{x=1}^{n} T_{GJx} \quad (8)$$

$$\text{where } T_{GJx} = T_{cx} + T_{ox} \quad (9)$$

n = number of job groups

$$T_{cx} / T_{ox} \geq 1 \quad (10)$$

Eq (6) reveals that the T_{ox} must not be more than T_{cx}

The above factors involved in job grouping technique offers minimum job processing time and maximum resource utilization of the Grid.

The steps involved in job grouping are as follows.

1) User creates a job list in the user machine.
2) The resource availability information is obtained from Grid Information Service (GIS).
3) The resources and jobs are sorted in descending order of their processing power and job length respectively.
4) From sorted list, the resources are selected one by one from reverse sorted resource list in first come first serve (FCFS) order.
5) Following the resource selection, jobs are added into GJ as per γ, BW and M by alternatively selecting jobs with maximum length from front end of the job list and jobs with minimum length from rear end of the job list. This is based on the following two conditions.

i. If Eq. (1) fails when adding a job into the group from front end of the job list,
Then
The respective job is removed from the group.
Preferably jobs are taken from the rear end of the job list till Eq. (1) gets satisfied.
End if
ii. If Eq. (1) fails when adding a job from rear end of the job list,
Then

Journal of Theoretical and Applied Information Technology
20th September 2014. Vol. 67 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Grouping of the respective resource is terminated removing the last job that was added and transmits the job group to the sender.
Next highest resource is considered to perform another job grouping.
End if
Note: The front end and rear end points are updated to represent the subsequent job in the list.
6) After grouping the jobs according to the resources available, the jobs are scheduled using enhanced genetic algorithm. (shown in section 3.4)

3.4.2 Job Scheduling

The proposed scheduling technique considers genetic algorithm (explained in section 3.2) which helps in minimizing the job completion time while performing certain execution of job within the application. The steps involved in the scheduling of job are as follows

1) Fitness function (F_i) is estimated (Shown in Eq (4)) based on the mobility, resource availability and job completion time and the estimated value is updated in the individual.
2) Based on F_i, the population transforms into the future generation.
3) To estimate the recombination and cross the individual, Roulette-Wheel selection technique is utilized. This involves the selection of chromosomes with higher F_i when compared to other chromosome for generating new offspring.
4) For F_i chromosomes, the selected probability σ_i is shown using Eq (11)

$$\sigma_i = \frac{F_i}{\sum F_i} \quad (11)$$

5) Two-point crossover is used to generate the new individuals by combining parents with certain estimated probability.
6) After the crossover, each bit of an individual is applied over the mutation operator.
7) The mutation operation replaces two randomly selected genes with the mutation probability P_m in the individual. The mutation enables offspring to provide better genes than its parents. This technique avoids duplication of individuals.
8) Thus the jobs are optimally scheduled using the global searching ability to improve the task analysis efficiency of mobile grid.

4. SIMULATION RESULTS

4.1 Simulation Model and Parameters

The Network Simulator (NS2) [17], is used to simulate the proposed architecture. In the simulation, 50 mobile nodes move in a 1000 meter x 1000 meter region for 50 seconds of simulation time. All nodes have the same transmission range of 250 meters. The simulated traffic is Constant Bit Rate (CBR).

The simulation settings and parameters are summarized in table.

No. of Nodes	50
Area Size	1000 X 1000
Mac	IEEE 802.11
Transmission Range	250m
Simulation Time	50 sec
Traffic Source	CBR
Packet Size	100
Speed	5,10,15,20 and 25m/s
Rate	10,20,30,40 and 50Kb

4.2 Performance Metrics

The proposed Improved Genetic Algorithm for Group-Based Job scheduling (IGAGJS) is compared with the TSGA technique [10]. The performance is evaluated mainly, according to the following metrics.

- **Packet Delivery Ratio:** It is the ratio between the number of packets received and the number of packets sent.
- **Packet Drop**: It refers the average number of packets dropped during the transmission
- **Throughput:** It is the number of packets received by the receiver during the transmission.
- **Delay**: It is the amount of time taken by the nodes to transmit the data packets.

4.3 Results

1) Based on Rate

In our first experiment we vary the rate as 10,20,30,40 and 50Kb.

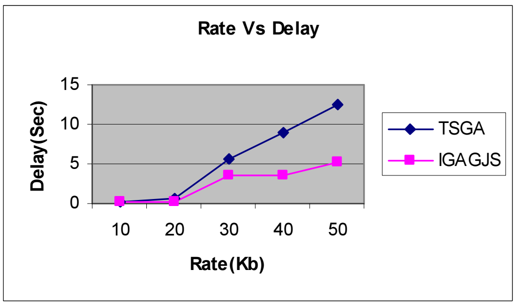

Fig 1: Rate Vs Delay

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

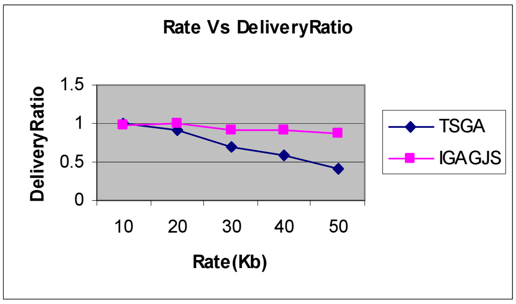

Fig 2: Rate Vs Delivery Ratio

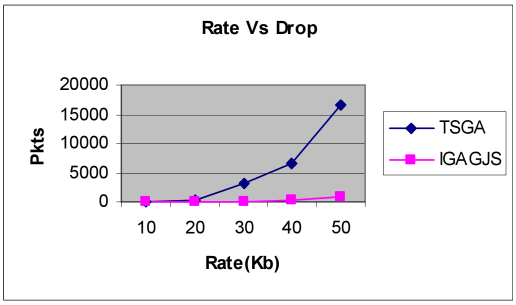

Fig 3: Rate Vs Drop

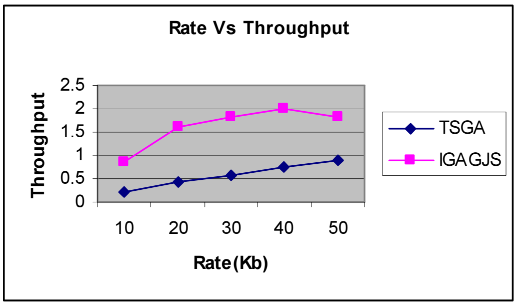

Fig 4: Rate Vs Throughput

Figure 1 shows the delay of IGAGJS and TSGA techniques for different rate scenario. We can conclude that the delay of our proposed IGAGJS approach has 44% of less than TSGA approach.

Figure 2 shows the delivery ratio of IGAGJS and TSGA techniques for different rate scenario. We can conclude that the delivery ratio of our proposed IGAGJS approach has 23% of higher than TSGA approach.

Figure 3 shows the drop of IGAGJS and TSGA techniques for different rate scenario. We can conclude that the drop of our proposed IGAGJS approach has 69% of less than TSGA approach.

Figure 4 shows the throughput of IGAGJS and TSGA techniques for different rate scenario. We can conclude that the throughput of our proposed IGAGJS approach has 66% of higher than TSGA approach.

2) Based on Speed

In our second experiment we vary the mobile speed as 5,10,15,20 and 25m/s

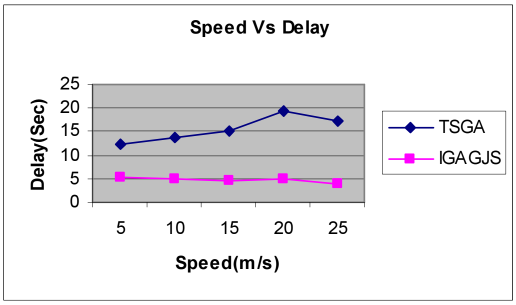

Fig 5: Speed Vs Delay

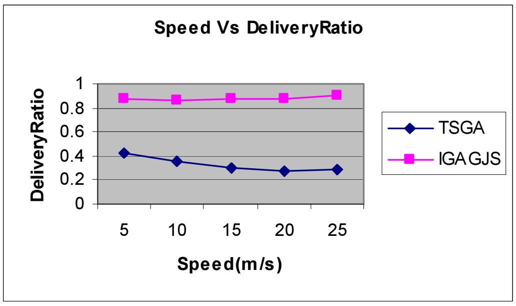

Fig 6: Speed Vs Delivery Ratio

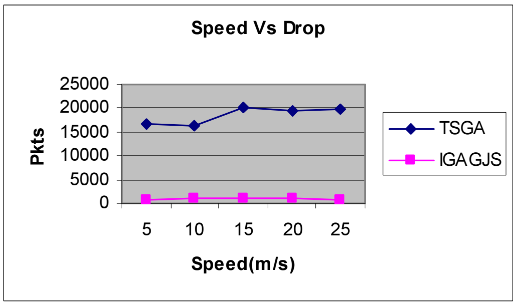

Fig 7: Speed Vs Drop

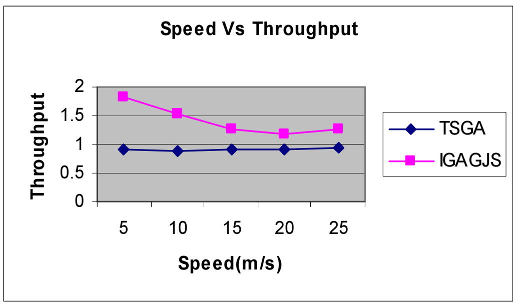

Fig 8: Speed Vs Throughput

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 5 shows the delay of IGAGJS and TSGA techniques for different speed scenario. We can conclude that the delay of our proposed IGAGJS approach has 68% of less than TSGA approach.

Figure 6 shows the delivery ratio of IGAGJS and TSGA techniques for different speed scenario. We can conclude that the delivery ratio of our proposed IGAGJS approach has 63% of higher than TSGA approach.

Figure 7 shows the drop of IGAGJS and TSGA techniques for different speed scenario. We can conclude that the drop of our proposed IGAGJS approach has 95% of less than TSGA approach.

Figure 8 shows the throughput of IGAGJS and TSGA techniques for different rate scenario. We can conclude that the throughput of our proposed IGAGJS approach has 34% of higher than TSGA approach.

5. CONCLUSION

In this paper, we have proposed an improved genetic algorithm for group-based job scheduling in mobile grids. In this technique, the jobs are grouped according to the resource availability. Then the jobs are scheduled based on the parameters such as mobility, resource availability and job completion time using enhanced genetic algorithm. By simulation results, we have shown that the proposed technique minimizes the job completion time, thus enhancing the system performance and minimizing the overhead time.

REFERENCES

[1] Antonios Litke, Dimitrios Skoutas, and Theodora Varvarigou, "Mobile grid computing: Changes and challenges of resource management in a mobile grid environment", *5th International Conference on Practical Aspects of Knowledge Management* (PAKM 2004). 2004.

[2] JongHyuk Lee, SungJin Song, JoonMin Gil, KwangSik Chung,Taeweon Suh and HeonChang Yu, "Balanced scheduling algorithm considering availability in mobile Grid", *Advances in Grid and Pervasive Computing,* Springer Berlin Heidelberg, 2009.

[3] Amir M Bidgoli and Zahra Masoudi Nezad, "A new scheduling algorithm design for grid computing tasks", *5th Symposium on Advances in Science and technology,* Iran 2010.

[4] S. Stephen Vaithiya and S. Mary Saira Bhanu, "Zone Based Job Scheduling In Mobile Grid Environment", *International Journal of Grid Computing & Applications (IJGCA),* June 2012.

[5] Liu, Lei, and Chunlin Li, "Mobile grid task scheduling considering resource reliability", *Computer Network and Multimedia Technology*, 2009, ©IEEE 2009.

[6] Birjea, M. N., and S. S. Manvi, "Agent Based Economic Scheme for Seamless Job Scheduling in Bandwidth Constrained Wireless Grids", *International Journal of Grid and Distributed Computing* 2012.

[7] Juan Manuel Rodriguez, Alejandro Zunino and Marcelo Campo, "Mobile grid seas: simple energy-aware scheduler", *Proc. 3rd High-Performance Computing Symposium-39th JAIIO.* 2010.

[8] Du Li-juan and Yu Zhen-wei, "Scheduling algorithm with respect to resource intermittence in mobile grid", *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on*, IEEE, 2010.

[9] Ashish Chandak, Bibhudatta Sahoo and Ashok Kumar Turuk, "Efficient Task Scheduling using Mobile Grid", ©2011 IEEE.

[10] Qing Jiang, Xianglin Wu and Haolan Yang, "Task Scheduling based on Genetic Algorithm in Mobile Grid", *2012 International Conference on Computer Science and Service System*, ©2012 IEEE.

[11] S. C Shah, S.H. Chaudhary, A. K. Bashirand M.S. Park, "A centralized location-based job scheduling algorithm for inter-dependent jobs in mobile ad hoc computational grids", *Journal of Applied Sciences*, 2010.

[12] Debabrata Singh, Sandeep Nanda, Sarbeswara Hota and Manas Kumar Nanda, "Evaluation of Scheduling and Load Balancing Techniques in Mobile Grid Architecture", *IJCSI International Journal of Computer Science Issues*,Vol. 8, September 2011.

[13] Sachin Kumar and Niraj Singhal, "A Study on the Assessment of Load Balancing Algorithms in Grid Based Network", *International Journal of Soft Computing and Engineering (IJSCE)* ISSN: 2231-2307, Volume-2, Issue-1, March 2012.

[14] Ali Sarhadi, Ali Yousefi and Ali Broumandnia, "New Method for Grid Computing Resource Discovery with Dynamic Structure of Peer-To-Peer Model Based on Learning Automata", *World Applied Sciences Journal* 19 (1): 153-158, 2012.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[15] Manoj Kumar Mishra, Prithviraj Mohanty and G.B Mund, "A Modified Grouping-Based Job Scheduling in Computational Grid", Engineering (NUiCONE), *2011 Nirma University International Conference on IEEE* 2011.

[16] JongHyuk Lee, SungJin Song, JoonMin Gil, KwangSik Chung, Taeweon Suh and HeonChang Yu, "Balanced Scheduling Algorithm Considering Availability in Mobile Grid", *Advances in Grid and Pervasive Computing*, Springer Berlin Heidelberg, 2009.

[17] Network Simulator: http:///www.isi.edu/nsnam/ns