Genetic Algorithm based Job Scheduling in Mobile Grids

G. Saravanan and V. Gopalakrishanan

Abstract--- *The existing job scheduling method increases the energy consumption, network performance and efficiency. In this paper, we have proposed a mobility based energy efficient job scheduling system using genetic algorithm (GA). At first based on the resource, availability jobs are grouped and assigned depends on subtasks and priorities. Then scheduling the jobs is developed using genetic algorithm based on the following parameters; mobility, resource availability, job completion time and energy. Finally, mobility prediction algorithm is used for estimating the accuracy of mobility. Hence, a simulation result is analyzed to show the effectiveness of the proposed system by minimizing the energy consumption and enhances the network performance.*

Keyword--- *Inertia, Position Updation, GA Operator, Velocity, Mobile Grid Computing.*

I. INTRODUCTION

A. Mobile Grids

A system that coordinates resources that are not subject to centralized control and that which uses standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of service is known as a Grid. The basis of Mobile Grid is the extension of the grid to mobile computing by making it accessible to the users even when they are mobile. Mobile Grid is relevant to both grid and mobile computing. It fully inherits grid, which has the additional feature of supporting mobile users and resources in an efficient way. The mobile grid is able to provide higher computational power and resources than the existing grid technology [1]

Mobile grid is helpful in the utilization of any unutilized resources on the devices. Apart from the above provision, the mobile grid can provide mobile devices with an opportunity to use the resources on the grid. Mobile devices having access to the grid as users can thus perform certain tasks on the run [2]

Grid computing use is based on the coordinated sharing of distributed and heterogeneous resources. This helps to resolve large-scale problems in dynamic virtual organizations. Mobile grid integrates traditional wired grid through wireless channel. Mobile devices have advantages over fixed computing resources. They are mobility, portability, and pervasiveness. The strengths inherent here allows mobile grid well applied to location-restricted fields requiring supportive infrastructure in wildfire prevention, disaster management and e-health system etc [3]

B. Mobility Prediction and Techniques in Mobile Grids

The main goal of mobility prediction is the facilitation of continuous access to grid resources. This is done irrespective of user's mobility. It also aims easy enabling of mobile devices to interact with resources in the grid. The most popular historical technique is Markov Models. Typically, a Markov mobility predictor performs the

G. Saravanan, Research Scholar Anna University. E-mail:gsaravanan.pacet@gmail.com
V. Gopalakrishanan, Associate Professor, Department of Electrical and Electronics Engineering, Government College of Technology, Thadagam Road, Coimbatore.

following operations. The first operation is maintaining a collection of past locations of the mobile users. The second operation is predicting future locations based on the value of conditional probability. This is done according to the match of the past locations of the mobile users [4] [5] [6].

There are many different mobility prediction proposed techniques .These techniques can be broadly classified into the following three categories: stochastic techniques, data mining techniques and pattern matching techniques A stochastic technique provides mobility predictions using probabilistic models. These techniques help to describe user's movements by assuming certain topographies of areas. Data mining techniques makes use of a database to track and characterize the long-term mobility patterns. These are further used to predict locations of mobile users. In pattern mining the learning process is dependent on the movement of an individual object available in a certain area [7].

C. Job Splitting for Mobile Grids

Jobs with a longer duration need to be split into a series of short jobs with the aid of check pointing. This helps to increase fault tolerance and to meet scheduling policy constraints of different resources. In order to take full advantage of grid environments, execution management systems is necessary to enable to configure, reconfigure, checkpoint and migrate jobs as necessary [8]

The process of job assignment or subtasks to the resources of a grid is known as scheduling, and this is executed by a grid scheduler. There are two classifications among the grid schedulers namely- Global Scheduler and Local Scheduler Ming and Sun (2003). The task scheduling is segmented into three parts, allocator, predictor and scheduler. The allocator decides how to allocate subtasks of a divisible application for each machine in a group of machines[9]

The rest of the paper is organized as follows. Section II provides various researches conducted in relation to our proposed work. Section III explains about the design strategy and the proposed method. Section IV shows the result and discussion of our proposed method and finally section V concludes method.

II. LITERATURE REVIEW

Einhorn et al., [10] have presented a novel mapping technique. It chooses the resolution of each cell adaptively by merging and splitting cells. This is dependent on the measurements. The splitting of the cells is based on a statistical measure. The derivation is shown in this paper. In contrast to other approaches the adaption of the resolution is done online, during the process of mapping itself.

Shah [11] have proposed a distributed computing infrastructure named mobile ad hoc computational Grid. This allows mobile nodes to share computing resources in mobile ad hoc environment. Compared to traditional parallel and distributed computing systems such as Grid and Cluster, mobile ad hoc is computational.

Miyashita et al., [12] have proposed a dynamic load distribution between computation nodes. This is done using mobile threads which lead to lightweight, low-overhead job relocation. Its effects are presented using an example problem, parallel Prefix Span which is used in the analysis of amino-acid sequences. The computation cost is absolutely unpredictable. However there is an occurrence of overhead migration.

Jeongy et al., [13] have proposed a new predictor. This utilizes paths instead of locations as the basis for prediction. Their path-based predictor makes a probabilistic prediction by extracting overlapping paths from past trajectories. This is done using a famous similarity measure, Frechet distance, there by constructing a tree based on junctions. A path is capable of inherent retaining correlation among locations; hence the proposed predictor is capable of achieving the ideal performance of order-1 Markov predictor with lower complexity.

Garg et al., [14] have presented a Meta scheduling algorithm. This exploits the heterogeneous nature of Grid which helps to achieve reduction in energy consumption. This algorithm significantly improves the energy efficiency of global grids by a factor of typically 23%. In some cases there is 50% while meeting user's QoS requirements. This influences the job urgency.

III. PROPOSED SOLUTION

A. Contribution of Our Work

In this paper, a mobility aware energy efficient job scheduling is proposed using genetic algorithm in mobile grids. Initially the jobs are grouped according to the resource availability. The grouped jobs are further split into sub-tasks and assigned priorities. Then the jobs are scheduled based on parameters like mobility, resource availability, job completion time and energy using enhanced genetic algorithm. Moreover, mobility prediction algorithm is used for estimating mobility accurately.

B. 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.

C. Estimation of Metrics

Let

T_u - predicted uptime.

T_d - predicted downtime.

T_c - time during which a network is connected.

T_{dc} - time during which a network is disconnected.

T_{ij} - 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): 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.

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

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)

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

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

Energy Consumption in Mobile Grid

In mobile grid, the major energy utility sources include computing devices (CPU) and cooling system. Other negligible sources include lighting etc.

The power consumption (E_{cpu}) of a CPU consists of dynamic and static power. The static power denotes the base power consumption of the CPU and the power consumption of all other components.

Let $Ncpu_i$ = number of CPU at resource

$$Ecpu = \alpha + \delta\tau^3$$

Where α = static power

δ = proportionality constant

f = frequency

Total energy consumption

$$Ec = \sum_{Ncpui}^{j} (\alpha + \delta\tau^3)\, tj \quad (4)$$

The energy cost of cooling system is based on co-efficient of performance (Ψ). Ψ is the ratio of amount of energy consumed by CPUs to energy consumed by cooling system. The total energy consumed by cooling system

$$Eco = Ec / \Psi \quad (5)$$

Thus total energy consumed by the grid resources $E_i = \left(1 + \frac{1}{\Psi}\right) E_c$

Fitness function: The fitness function of the chromosome is computed based on the mobility, resource availability and job completion time using Eq. (6).

$$Fi = \beta_1 (TJC) + \beta_2 (RA) + \beta_3 (MOi) + \beta_4 (E_i) \quad (6)$$

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

D. Mobility Aware Energy Efficient Job Scheduling

Our proposed technique involves two phases

Phase I: Grouping of Jobs

Phase II: Job Splitting

Grouping of Jobs

Let

GJ be the grouped job

γ - resource's processing capability measured in MIPS Million Instruction per Second.

P - computational power required by the job measured using Million Instructions.

t - user defined time used to measure total amount of GJ completed within a specific time.

MJ - memory size of GJ

M - memory available at resources

BW - bandwidth capacity of the resources

t_c - communication time of the time

T_{GJx} - processing time of x^{th} GJ

T_{ox} - overhead time of x^{th} GJ

T_{cx} - 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.

$$P \leq \gamma * t \quad (7)$$

$$MJ \leq M \quad (8)$$

$$MJ \leq BW * tc \quad (9)$$

Eq. (7) reveals that P should not exceed γ.

Eq. (8) reveals MJ should not exceed M.

Eq. (9) reveals that MJ should not exceed to BW within specific time period.

The job completion time (T_{JC}) is estimated using the following Eq (10)

$$TJC = \sum_{x=1}^{n} T_{GJx} \quad (10)$$

$$\text{where } T_{GJx} = T_{cx} + T_{ox} \quad (11)$$

n = number of job groups

$$T_{cx} / T_{ox} \geq 1 \quad (12)$$

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.

Job Splitting Based on Grid Harvest Service

The jobs which are grouped as per resource availability are partitioned into subtasks using grid harvest service technique. The job splitting decision is performed using task assigner and partitioned jobs are mapped to set of resources using mean-time allocation algorithm.

Let

r - Resource μ_r - Resource utilization

P_r - processing power $r_1, r_2, ..., r_n$ - list of resources J - Workload

The mapping the sub-task (m_r) to each resource $r_1, r_2, ..., r_n$, is performed using following Eq

Begin

For each machine r_n ($1 \le r \le n$)

$$m_r = \frac{m}{\sum_{r=1}^{n}(1-\mu_r)P_r} \quad (13)$$

End for

Return $m_r(1 \le r \le n)$

End

Prioritizing the Partitioned Sub-Tasks

The partitioned sub-tasks are classified into high, low and medium categories as per the priority using Medium Subtask Fastest Node algorithm (MidSFN). The priority is assigned to each job based on the computational complexity (CC) of the sub-task and performance factor (PF) of the resource. PF is estimated using the following Eq

Let

N_o = number of operations per cycle per processor N_{IP} = number of instructions processed per second (processor speed)

LL = local list of nodes available in grid resource. Pr (i) = priority

m = number of free nodes available in the resource. n = number of subtasks of a job presents in the queue.

$$PF = N_o * N_{IP} \quad (14)$$

The resultant PFL (RPFL) contains the performance factor computed and sorted in descending order for each node available in the resource.

If (CC (i) = medium) & (RPFL(i) = medium) Then Pr(i) = 1

Else If (CC(i) = High AND RPFL(i) = Medium) Then Pr(i) = 2

Else If (CC(i)= Low AND RPFL(i) =Medium) Then Pr(i) = 3

Else If (CC(i)= Medium AND RPFL(i) = High) Then Pr(i) = 4

Else If (CC(i)= High AND RPFL(i) = High) Then Pr(i) = 5

Else If (CC(i)= Low AND RPFL(i) = High) Then Pr(i) = 6

Else If (CC(i)= Medium AND RPFL(i) = Low) Then Pr(i) = 7

Else If CC(i) = High AND RPFL(i) = Low) Then Pr(i) = 8

Else If CC(i)= Low AND RPFL(i) = Low) Then Pr(i) = 9

End if

$$\text{The worst case time complexity of above algorithm} = \begin{cases} O\,(n \log n)\,, & m \le n \\ O\,(m \log m), & m > n \end{cases} \tag{15}$$

Job Scheduling

The proposed scheduling technique considers genetic algorithm. This is explained in section 3.2) It helps to minimize the job completion time while performing certain execution of jobs within the application. The steps involved in the scheduling of job are as follows:

1) Fitness function (F_i) is estimated (Shown in Eq (6)) based on the mobility, resource availability, job completion time and energy 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 (16)

$$\sigma_i = \frac{F_i}{\sum F_i} \tag{16}$$

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 .It has the mutation probability P_m in the individual. The mutation enables offspring to provide better genes than its parents. The technique thus helps to avoid duplication of individuals
8) Thus there is optimal scheduling of jobs using the global searching ability to improve the task analysis efficiency of mobile grid.

Mobility Prediction

The mobility prediction algorithm uses straight forward mobility predictor .It involves two steps- Pre-processing and Definite prediction Preprocessing

1) The future and past location of node is estimated and recorded in a file initially. This is performed using global positioning system.

2) The recorded large file is broken into separate files for each user.
3) The access point records sensed by the devices that are not related are dropped.
4) The continuous records are merged into sessions. where each session consists of starting time, related access point and session duration.

Definite Prediction

The location obtained in pre-processing step is considered as input in prediction step. The prediction technique involves following steps.

1) A set (Q_i) of node with past locations (G_i) and current location C_i is constructed.
2) Q_i is compared with history sequence (H_i) that includes G_i, C_i and subsequent n location.
3) The result sequence is added into the prediction list PL.
4) If PL = NULL Then The session with minimum duration is removed from G_i
   End if The search of H_i is repeated until PL contains at least one historical sequence.
5) The subsequent n location is predicted by estimating the probability for each of the resultant locations appearing in the history set.

IV. SIMULATION RESULTS

A. Simulation Model and Parameters

The Network Simulator (NS2) http:///www.isi.edu/nsnam/ns, 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
Initial Energy	14.3J
Receiving Power	0.395
Transmission Power	0.660

B. Performance Metrics

The proposed Mobility Aware Energy Efficient Job Scheduling (MAEEJS) is compared with Improved Genetic Algorithm for Group-Based Job scheduling (IGAGJS) Kołodziej et al., (2012). The performance is evaluated mainly, according to the following metrics.

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

C. Results

1) Based on Rate

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

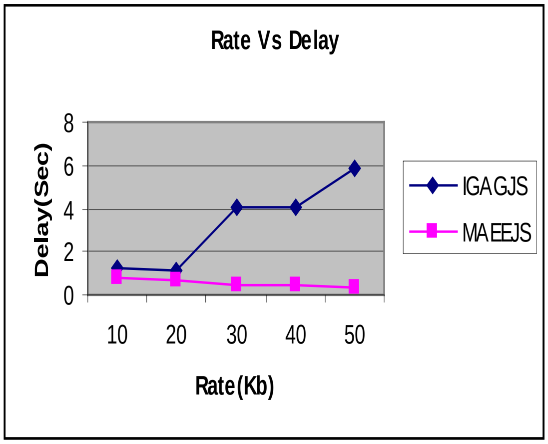

Figure 1: Rate Vs Delay

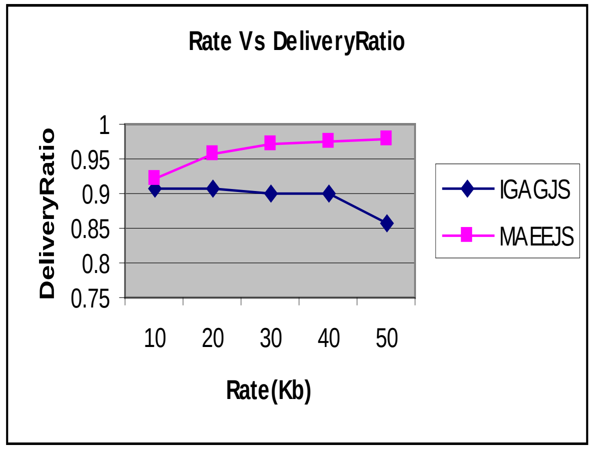

Figure 2: Rate Vs Delivery Ratio

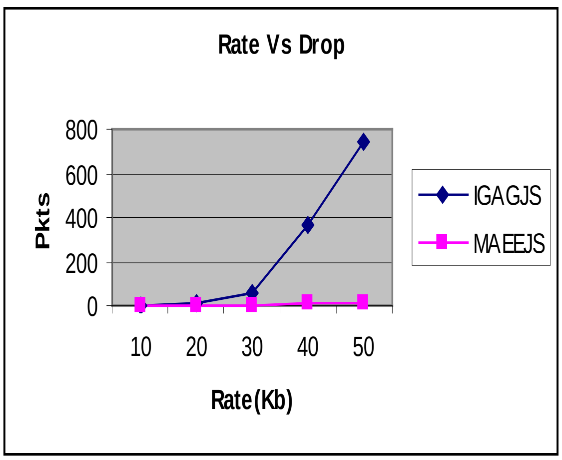

Figure 3: Rate Vs Drop

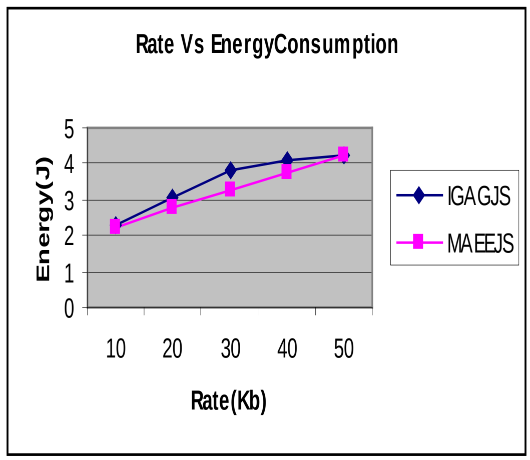

Figure 4: Rate Vs Energy Consumption

Figure 1 shows the delay scenario. We can conclude that the delay of our proposed MAEEJS approach has 69% of less than IGAGJS approach.

Figure 2 shows the delivery ratio scenario. We can conclude that the delivery ratio of our proposed MAEEJS approach has 7% of higher than IGAGJS approach.

Figure 3 shows the drop scenario. We can conclude that the drop of our proposed MAEEJS approach has 66% of less than IGAGJS approach.

Figure 4 shows the energy consumption scenario. We can conclude that the energy consumption of our proposed MAEEJS approach has 7% of less than IGAGJS approach.

2) Based on Speed

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

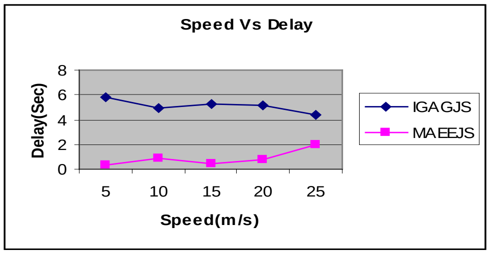

Figure 5: Speed Vs Delay

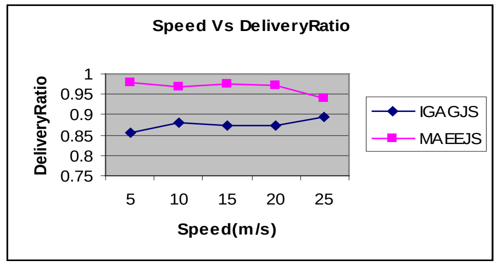

Figure 6: Speed Vs Delivery Ratio

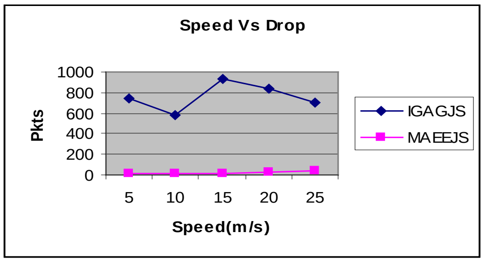

Figure 7: Speed Vs Drop

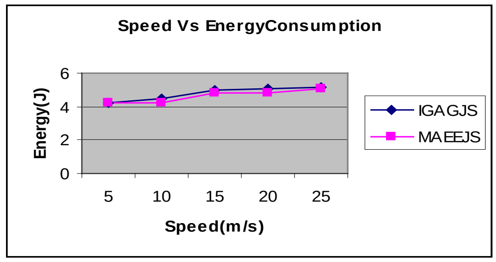

Figure 8: Speed Vs Energy Consumption

Figure 5 shows the delay scenario. We can conclude that the delay of our proposed MAEEJS approach has 81% of less than IGAGJS approach.

Figure 6 shows the delivery scenario. We can conclude that the delivery ratio of our proposed MAEEJS approach has 9% of higher than IGAGJS approach.

Figure 7 shows the drop scenario. We can conclude that the drop of our proposed MAEEJS approach has 96% of less than IGAGJS approach.

Figure 8 shows the energy scenario. We can conclude that the energy consumption of our proposed MAEEJS approach has 3% of less than IGAGJS approach.

V. CONCLUSION

In this paper, we have proposed a mobility aware energy efficient job scheduling. This is done using genetic algorithm in mobile grids. The initial grouping of jobs is done according to the resource availability. The grouped jobs are split into sub-tasks and priorities are assigned. Then the jobs are scheduled taking into consideration the parameters like mobility, resource availability, job completion time and energy using enhanced genetic algorithm. A mobility prediction algorithm is used for the accurate estimation of mobility. By simulation results, we have proved that the proposed technique minimizes energy consumption, thereby enhancing the network performance. In addition to this as a future work, we can improve the system by using hybridization techniques such as genetic algorithm will be combined with one other algorithm to improve the performance and metrics will be increased to show the performance evaluation

REFERENCES

[1] G. Saravanan and V. Gopalakrishanan, "Mobility Aware Energy Efficient Job Scheduling Using Genetic Algorithm in Mobile Grids", European Journal of Applied Sciences, Vol. 7, No. 3, Pp. 117-125, 2015.
[2] A. Bichhawat and R.C. Joshi, "Proactive Fault Tolerance Technique for a Mobile Grid Environment".
[3] S.S. Vaithiya and S.M.S. Bhanu, "Zone Based Job Scheduling in Mobile Grid Environment", International Journal of Grid Computing & Applications (IJGCA), Vol. 3, No. 2, 2013.
[4] W. Ming and X.H. Sun, "A general self-adaptive task scheduling system for non-dedicated heterogeneous computing", International Conference on Cluster Computing, Pp. 354-361, 2003.
[5] X. Sun and M. He, "Grid Harvest Service: a system for long-term, application-level task scheduling", Proceedings Parallel and Distributed Processing Symposium, 2003.
[6] M. Ivanovic, V. Simic and B. Stojanovic, "Elastic grid resource provisioning with WoBinGO: A parallel framework for genetic algorithm based optimization", Future Generation Computer Systems, Vol. 42, Pp. 44-54, 2015.
[7] I. Khalifa and H.M. Abbas, "Mobility Prediction in Dynamic Grids", Computer & Information Science, Vol. 5, No. 3, 2012.
[8] T. Elahi, C. Kiddle and R. Simmonds, "Models for Grid Applications and Jobs", 22nd International Symposium on High Performance Computing Systems and Applications, Pp. 52-58, 2008.
[9] Network Simulator: http:///www.isi.edu/nsnam/ns
[10] E. Einhorn, C. Schroter and H.M. Gross, "Finding the Adequate Resolution for Grid Mapping - Cell Sizes Locally Adapting On the Fly", IEEE International Conference on Robotics and Automation Shanghai International Conference Center, Pp. 1843-1848, 2011.
[11] S.C. Shah, "Mobile Ad hoc Computational Grid:Opportunities and Challenges", Military Communications, Pp. 848-857, 2013.
[12] M. Miyashita, E. Haque, N. Matsumoto and N. Yoshida, "Dynamic Load Distribution in Grid Using Mobile Threads", International Conference on High Performance Computing and Communications (HPCC), Pp. 629-634, 2010.
[13] J. Jeongy, K. Leez, K. Leey and S. Chong, "Path-based Mobility Prediction: Breaking through the Limit of Location-based Prediction", Technical Report.
[14] S.K. Garg and R. Buyya, "Exploiting heterogeneity in Grid computing for energy-efficient resource allocation", 17th International Conference on Advanced Computing and Communications, 2009.