European Journal of Applied Sciences 7 (3): 117-125, 2015
ISSN 2079-2077

DOI: 10.5829/idosi.ejas.2015.7.3.22240

Mobility Aware Energy Efficient Job Scheduling Using Genetic Algorithm in Mobile Grids

[1]G. Saravanan and [2]V. Gopalakrishanan

[1]Computer Science and Engineering Excel Engineering College Salem Main Road,
NH 47, Pallakapalayam, Komarapalayam - 637 303, India
[2]Departmnet of Electrical and Electronics Engineering Government College of Technology,
Thadagam Road,Coimbatore - 641013, India

Abstract: In mobile grids, the existing job scheduling scheme causes increased energy consumption. Also there is reduced network performance and efficiency. Hence in this paper, we propose a mobility aware energy efficient job scheduling using genetic algorithm in mobile grids. Initially the jobs are grouped according to the resource availability. The grouped jobs are split into sub-tasks and priorities are assigned. Then the jobs are scheduled based on the parameters such as mobility, resource availability, job completion time and energy using enhanced genetic algorithm. A mobility prediction algorithm is used for estimating mobility accurately. By simulation results, we show that the proposed technique minimizes the energy consumption and enhances the network performance.

Key words: Grid computing · Portability · Computational Complexity · Path-based predictor

INTRODUCTION

Mobile Grids: A Grid is defined as a system that coordinates resources that are not subject to centralized control which uses standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of service. The extension of the grid to mobile computing by making it available to the users even when they are mobile forms the basis of Mobile Grid. Mobile Grid, in relevance to both grid and mobile computing, is a full inheritor of grid with the additional feature of supporting mobile users and resources in efficient way. The mobile grid can provide higher computational power and resources than the existing grid technology.

Mobile grid could help in the utilization of any unutilized resources on the devices. Apart from utilizing resources on the mobile devices, the mobile grid can provide mobile devices with an opportunity to use the resources on the grid, thereby saving their own resources considerably and overcoming the physical shortcoming of the device. Mobile devices having access to the grid as users would thus be able to perform certain tasks on the run which otherwise would have been done only when the user could access a wired device.

Grid computing is used based on the coordinated sharing of distributed and heterogeneous resources to solve large-scale problems in dynamic virtual organizations. Mobile grid integrates traditional wired grid through wireless channel to share grid resources to mobile users or provide resources to grid. Mobile devices have advantages over fixed computing resources such as mobility, portability and pervasiveness. These strengths allow mobile grid well-applied to location-restricted fields requiring supportive infrastructure in wildfire prevention, disaster management and e-health system etc.

Mobility Prediction and Techniques in Mobile Grids: The main goal of mobility prediction is to facilitate continuous access to grid resources irrespective of user's mobility. It also aims to enable mobile devices to easily interact with resources in the grid. The most popular history based technique is Markov Models. Typically, a Markov mobility predictor performs the following two operations, the first operation is to maintain a collection of past locations of the mobile users, while the second operation is to predict future locations based on the value of conditional probability that matches the past locations of the mobile users.

Corresponding Author: G. Saravanan, Computer Science and Engineering Excel Engineering College Salem Main Road, NH 47, Pallakapalayam, Komarapalayam - 637 303, India.

Many different mobility prediction techniques have been proposed, these techniques can be broadly classified into the following three categories:

- Stochastic techniques
- Data mining techniques
- Pattern matching techniques

Stochastic techniques provide mobility predictions using probabilistic models. These techniques provide means to describe user's movements by assuming certain topographies of areas. Data mining techniques uses a database to track and characterize the long-term mobility patterns, which are then used to predict locations of mobile users. In pattern mining the learning process depends on the movement of an individual object available in a certain area.

Job Splitting for Mobile Grids: Long running jobs need to be split into a series of shorter jobs with the aid of check pointing to increase fault tolerance and to meet scheduling policy constraints of different resources. To take full advantage of grid environments, execution management systems need to be able to configure, reconfigure, checkpoint and migrate jobs as necessary.

The process of assigning jobs or subtasks to the resources of a grid is known as scheduling and this is done by a grid scheduler. There are two classifications among the grid schedulers Global Scheduler and Local Scheduler. The task scheduling is divided into three parts, allocator, predictor and scheduler. Allocator decides how to allocate subtasks of a divisible application for each machine in a group of machines, where a divisible application refers to an application that could be partitioned into a set of subtasks. Predictor estimates the application execution time distribution on each machine. Scheduler decides which set of machines is the best among all sets of machines.

Literature Review: Erik Einhorn *et al*. [1] have presented a novel mapping technique that chooses the resolution of each cell adaptively by merging and splitting cells depending on the measurements. The splitting of the cells is based on a statistical measure that is derived in this paper. In contrast to other approaches the adaption of the resolution is done online during the mapping process itself. Additionally, they introduced the Nd-Tree, a generalization of quadtrees and octrees that allows subdividing any d-dimensional volume recursively with Nd children per node. Using this data structure their approach can be implemented in a very generic way and allows the creation of 2D, 3D and even higher dimensional maps using the same algorithm. However there occurs loss of information.

Sayed Chhattan Shah [2] have proposed a distributed computing infrastructure named mobile ad hoc computational Grid which 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 computational Grid is characterized by shared and unreliable communication medium, low bandwidth, high latency, node mobility and infrastructure-less network environment, so it introduces numerous opportunities as well as challenges. For resource allocation three schemes where implemented POW-TPRA, NLRA and DRA. However POW-TPRA does not perform well due to the transmission power control mechanism which generates large amount of computational overhead.

Masaya Miyashita *et al*. [3] have proposed a dynamic load distribution between computation nodes using mobile threads which lead to lightweight, low-overhead job relocation. They also have presented its effects using an example problem, parallel Prefix Span which is used in the analysis of amino-acid sequences, whose computation cost is absolutely unpredictable. However there occurs migration overhead.

Jaeseong Jeongy *et al.* [4] proposed a new predictor, which utilizes paths instead of locations as the basis for prediction. Their path-based predictor probabilistically makes a prediction by extracting overlapping paths from past trajectories using a famous similarity measure, Frechet distance and constructing a tree based on junctions. Since a path inherently retains correlation among locations, the proposed predictor is capable of achieving the ideal performance of order-1 Markov predictor with lower complexity. The results in real dataset show that path-based prediction has significant improvement in both predictability and precision. However there is complexity in the proposed method.

Saurabh Kumar Garg and Rajkumar Buyya [5] have presented a Meta scheduling algorithm which exploits the heterogeneous nature of Grid to achieve reduction in energy consumption. This algorithm can significantly improve the energy efficiency of global grids by a factor of typically 23% and as much as a factor of 50% in some cases while meeting user's QoS requirements. However the job urgency is influenced.

Joanna Kołodziej *et al*. [6] have presented the application of GAs for solving the energy-aware scheduling problem in computational grids. The energy consumption management model is based on DVFS technique adapted to the dynamic grid environment. They formalized the grid scheduling problem as a bi-objective optimization task with Make span and Average energy consumption as the main objectives. Also they developed three versions of the energy aware genetic-based metaheuristics with different replacement mechanisms, namely Steady-state GA (GA-SS), GA with Elitist generational replacement (GA-EG) and Struggle GA (GA-ST) for solving the considered grid scheduling problem. The results confirmed the effectiveness of the proposed genetic algorithm-based schedulers in the reduction of the energy consumed by the whole system and in dynamic load balancing of the resources in grid clusters, which is sufficient to maintain the desired quality level. However it has been observed that the skewness of the distribution of the results is positive or neutral for the worst "energy optimizers" and negative for the best ones.

Dr. G. Sumathi *et al.* [7] have presented Medium Subtask Fastest Node algorithm which classifies the subtasks into three tier categories, High, Middle and Low based on their priority. In MidSFN algorithm priority is assigned based on the new parameters Computational Complexity and Processing Power. The value for processing power is assigned based on the Performance Factor. The value of the Performance Factor is the product of the number of operations per cycle per processor and the number of instructions processed per second. In MidSFN algorithm the subtask of medium computational complexity and resources exhibiting medium processing power are assigned with a high priority. The subtasks are then mapped to respective processors based on the assigned priority for execution. Compared to other local scheduling algorithms, MidSFN algorithm shows efficient load balancing and better computation with effective usage of resources.

Proposed Solution

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

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

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.

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

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.

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[9-12].

Let $Ncpu_i$ = number of CPU at resource

$$E_{cpu} = \alpha + \delta\tau^3$$

where,
α = static power
δ = proportionality constant
f = frequency
Total energy consumption

$$E_c = \sum_{Ncpui}^{j} (\alpha + \delta\tau^3) t_j \quad (4)$$

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 [13]. The total energy consumed by cooling system.

$$E_{co} = E_c / \Psi \quad (5)$$

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).

$$F_i = \beta_1 (T_{JC}) + \beta_2 (RA) + \beta_3 (MO_i) + \beta_4 (E_i) \quad (6)$$

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

Mobility Aware Energy Efficient Job Scheduling:

Our proposed technique involves two phases
Phase I: Grouping of Jobs
Phase II: Job Splitting
Phase III:

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[14].
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.

$$P \leq \gamma * t \quad (7)$$
$$MJ \leq M \quad (8)$$
$$MJ \leq BW * t_c \quad (9)$$

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 (10).

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

where $T_{GJx} = T_{cx} + T_{ox}$ (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.

- User creates a job list in the user machine.
- The resource availability information is obtained from Grid Information Service (GIS).
- The resources and jobs are sorted in descending order of their processing power and job length respectively.
- From sorted list, the resources are selected one by one from reverse sorted resource list in first come first serve (FCFS) order.
- 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.
- 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

- If Eq. (1) fails when adding a job from rear end of the job list,

Then

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.

- After grouping the jobs according to the resources availability, the jobs are split into sub-tasks using Grid Harvest Service. (shown in section 3.4)

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.

The job is assigned to each resource as per the expected mean execution time of subtasks on the resource.

Let r be the resource
Let γ be the resource utilization
Let P_r be the processing power
Let $r_1, r_2, \ldots, r_n$ be the list of resources
Let J be the workload

The mapping the sub-task (m_r) to each resource $r_1, r_2, \ldots, r_n$, 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
Let N_{IP} = number of instructions processed per second (processor speed)
Let LL = local list of nodes available in grid resource.
Let Pr (i) = priority
Let m = number of free nodes available in the resource.
Let 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
```

The worst case time complexity of above algorithm

$$= \begin{cases} O(n \log n), & m \le n \\ O(m \log m), & m > n \end{cases} \quad (15)$$

Thus, our proposed algorithm offers higher priority for the sub-tasks with medium processing power and medium computational complexity. i.e., the fastest node available in the resources is allocated with high priority. This technique optimizes the computational speed of the grid and reduces the usage of nodes and also offers consistent performance during execution of the assigned subtask.

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

- 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.
- Based on F_i, the population transforms into the future generation.
- 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.
- For F_i chromosomes, the selected probability σ_i is shown using Eq. (16)

$$\sigma_i = \frac{F_i}{\sum F_i} \quad (16)$$

- Two-point crossover is used to generate the new individuals by combining parents with certain estimated probability.
- After the crossover, each bit of an individual is applied over the mutation operator.
- 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.
- Thus the jobs are optimally scheduled 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 and it involves following two steps. Pre-processing and Definite prediction

Preprocessing:

- Initially, the future and past location of node is estimated and recorded in a file. This is performed using global positioning system.
- For each user, the recorded large file is broken into separate files.
- The records where the access point has been sensed by the devices, however it is not related, is dropped.
- The continuous records are merged into sessions. Each session includes 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.

- A set (Q_i) of node with past locations (G_i) and current location C_i is constructed.
- Q_i is compared with history sequence $(H)_i$ that includes G_i, C_i and subsequent n location.

- The result sequence is added into the prediction list PL.
- 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.

- The subsequent n location is predicted by estimating the probability for each of the resultant locations appearing in the history set.

Simulation Results

Simulation Model and Parameters: The Network Simulator (NS2) [14], 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 1.

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

Performance Metrics: The proposed Mobility Aware Energy Efficient Job Scheduling (MAEEJS) is compared with Improved Genetic Algorithm for Group-Based Job scheduling (IGAGJS) [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.

RESULTS

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

Figure 1 shows the delay of MAEEJS and IGAGJS techniques for different rate 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 of MAEEJS and IGAGJS techniques for different rate 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 of MAEEJS and IGAGJS techniques for different rate scenario. We can conclude that the drop of our proposed MAEEJS approach has 66% of less than IGAGJS approach.

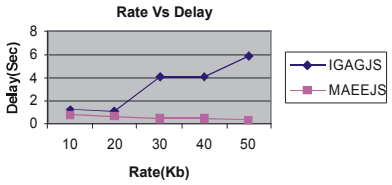

Fig. 1: Rate Vs Delay

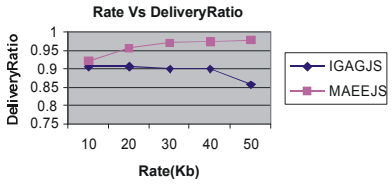

Fig. 2: Rate Vs Delivery Ratio

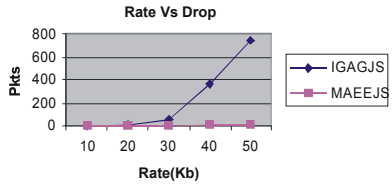

Fig. 3: Rate Vs Drop

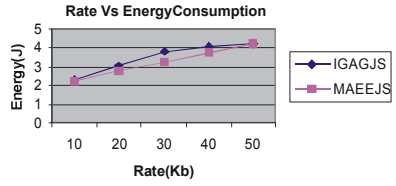

Fig. 4: Rate Vs Energy Consumption

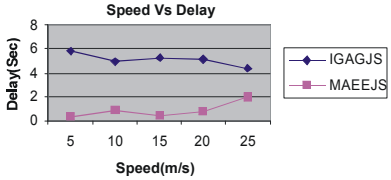

Fig. 5: Speed Vs Delay

Fig. 6: Speed Vs Delivery Ratio

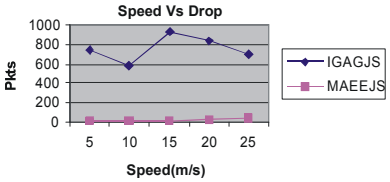

Fig. 7: Speed Vs Drop

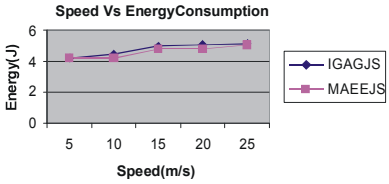

Fig 8: Speed Vs Energy Consumption

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

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

Figure 1 shows the delay of MAEEJS and IGAGJS techniques for different speed scenario. We can conclude that the delay of our proposed MAEEJS approach has 81% of less than IGAGJS approach.

Figure 2 shows the delivery ratio of MAEEJS and IGAGJS techniques for different speed scenario. We can conclude that the delivery ratio of our proposed MAEEJS approach has 9% of higher than IGAGJS approach.

Figure 3 shows the drop of MAEEJS and IGAGJS techniques for different speed scenario. We can conclude that the drop of our proposed MAEEJS approach has 96% of less than IGAGJS approach.

Figure 4 shows the energy consumption of MAEEJS and IGAGJS techniques for different speed scenario. We can conclude that the energy consumption of our proposed MAEEJS approach has 3% of less than IGAGJS approach.

CONCLUSION

In this paper, we have proposed a mobility aware energy efficient job scheduling using genetic algorithm in mobile grids. Initially the jobs are grouped according to the resource availability. The grouped jobs are split into sub-tasks and priorities are assigned. Then the jobs are scheduled based on the parameters such as mobility, resource availability, job completion time and energy using enhanced genetic algorithm. A mobility prediction algorithm is used for estimating mobility accurately. By simulation results, we have shown that the proposed technique minimizes the energy consumption and enhances the network performance.

REFERENCES

1. Einhorn Erik and Christof Schroter and Horst-Michael Gross, 2011. 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 May 9-13, 2011.
2. Shah Sayed Chhattan, 2013. Mobile Ad hoc Computational Grid: Opportunities and Challenges, Military Communications Conference, MILCOM 2013, IEEE.
3. Miyashita Masaya, Md. Enamul Haque, Noriko Matsumoto and Norihiko Yoshida, 2010. Dynamic Load Distribution in Grid Using Mobile Threads, High Performance Computing and Communications (HPCC), 2010 12th IEEE International Conference on. IEEE.
4. Jeongy Jaeseong, Kyunghan Leez, Kimin Leey and Song Chong, Path-based Mobility Prediction: Breaking through the Limit of Location-based Prediction -Technical Report.
5. Garg, Saurabh Kumar and Rajkumar Buyya, 2009. Exploiting heterogeneity in Grid computing for energy-efficient resource allocation, Proceedings of the 17th International Conference on Advanced Computing and Communications (ADCOM 2009), Bengaluru, India.

6. Kołodziej Joanna, Samee Ullah Khan, Lizhe Wang and Albert Y. Zomaya, 2012. Energy efficient genetic-based schedulers in computational grids, Concurrency and Computation: Practice and Experience.
7. Sumathi G., R. Santhosh Kumar and S. Sathyanarayanan, 2012. MidSFN Local Scheduling Algorithm for Heterogeneous Grid Environment, IJCSI International Journal of Computer Science, 9(3): 3.
8. Sun, Xian-He and Ming Wu, 2003. Grid Harvest Service: a system for long-term, application-level task scheduling, Parallel and Distributed Processing Symposium, 2003. Proceedings. International. IEEE.
9. Ming, Wu and Xian-He Sun, 2003. A general self-adaptive task scheduling system for non-dedicated heterogeneous computing, Cluster Computing, 2003. Proceedings 2003 IEEE International Conference on. IEEE.
10. Network Simulator: http:///www.isi.edu/nsnam/ns.
11. Bichhawat, Abhishek and R.C. Joshi, 2012. Proactive Fault Tolerance Technique for a Mobile Grid Environment.
12. Stephen, Vaithiya S. and S. Mary Saira Bhanu, 2012. Zone Based Job Scheduling in Mobile Grid Environment, International Journal of Grid Computing & Applications (IJGCA), 3(2).
13. Iraky, Khalifa and Hala Mohamed Abbas, 2012. Mobility Prediction in Dynamic Grids, Computer & Information Science 5.3.
14. Elahi Tanvire, Cameron Kiddle and Rob Simmonds, 2008. Models for Grid Applications and Jobs, 22nd International Symposium on High Performance Computing Systems and Applications (HPCS).