Maximizing Reliability with Task Scheduling In a Computational Grid Using GA

Zahid Raza *, Deo Prakash Vidyarthi

School of Computer and Systems Sciences
Jawaharlal Nehru University
New Delhi-110067
{zahidraza, dpv}@mail.jnu.ac.in
doi: 10.4156/ijact.vol1.issue2.5

Abstract

Grid is a service aggregation of both the information and the computational resources. Scheduling becomes a challenging job in such a complex and dynamic environment as both the application and the computational resources are heterogeneous. The problem is further complicated by the fact that these resources may fail at any point of time. Thus a scheduling strategy which schedules the job based on the failure possibility of the grid constituents becomes very important for the reliable execution of the job. Genetic algorithm has evolved as an effective search tool to solve the optimization problems consisting of large search space. It is a type of evolutionary search strategies applied on the search space and is based on the principle of “survival of the fittest”. The model addresses the important issue of scheduling to provide the most reliable environment for the job execution by scheduling the job, ensuring maximum reliability to the job execution using Genetic Algorithm. The model presents a realistic picture of the grid by scheduling the job based on the reliability of the computational resources, networking resources and application (job) along with the pre-assigned workload on the nodes in order to provide the most reliable environment to the job. Simulation study proves the effectiveness of the model by comparing the performance of the model with another similar model.

Keywords
Computational Grid, Scheduling, Reliability, Genetic Algorithm (GA), NP problems.

1. Introduction

The objective of the Grid is to utilize all available resources that are under utilized, to the maximum possible extent. It happens quite often, in any organization, that there is huge computing power available cumulatively but only a little of it is utilized to perform the computing work. Also, sometimes a job which requires high computing energy takes substantive time in producing the results due to the non availability of the computing energy. Further, sometimes the job may need a specialized computing platform which may not be available at the node where the job is submitted. A possible solution is to integrate the computing resources available at various places and enable all authorized users, to use it in a well-defined transparent manner. The hunger for the enormous computational power by various applications (e.g. weather forecasting, molecular study, numerical data processing, particle physics etc.) has led the scientific community towards the computational power sharing and cooperation, resulting in a computational grid. Grid enables the use of networks of computers as a single, unified computing tool, clustering and coupling a variety of facilities over a wide geographic region [6-8, 17].

Scheduling a job on a grid is an NP problem. Therefore, a number of models have been proposed in the literature with a scheduling strategy with one or the other objectives. These objectives are the Job completion time (Turn around time), Fault tolerance, Reliability, Quality of service (QoS), Real time completion, Resource utilization, Security etc. The proposed study is based on scheduling the job submitted for execution on a grid with the objective to maximize the reliability of the job execution. This is done by scheduling the job over the reliable resources at any given moment of time.

A genetic algorithm is an effective tool to solve NP class of problems. It works on the basis of natural selection of the fittest, among the population of individuals, according to a fitness function and then evolving towards a definite goal over the generations. In each generation the individuals undergo crossover to reproduce better off-springs. By repeating the process over generations, it gradually tends to the optimization of the desired parameter. Mutation is used to avoid local optima during the optimization process to avoid false
peaks. The results obtained from GA may not be the best solution but certainly is better till that generation. GA is different from the classic optimization techniques as it works on a set of points in the search space rather than a single point. It uses probabilistic approach in selection rather than deterministic one and also uses the knowledge gained from the past history in generating future strings [9-12].

Section 2 discusses the reliability and scheduling requirements for the grid, section 3 elaborates the proposed model whereas section 4 explains the working of the proposed model and the algorithm. Section 5 discusses the results obtained from the simulation study. The paper ends in Section 6 with concluding remarks.

2. Scheduling and Reliability for a Grid

Whenever a task enters the grid for execution the chances for failure may spread from the application failure to the resource failure [10]. In such a hostile environment, it is always desired to schedule the job in the most reliable way. The proposed model schedules the job, awaiting execution, to the most reliable resources at that point of time. This results in the reliable job execution increasing the probability of its successful completion, as required.

Most of the work reported in the literature lags in providing the most reliable execution environment to the job. The reliability estimations are done over the distributed computing system with one or the other parameters like reliability of the application, nodes or the distribution network [4, 13, 19]. Majority of the models do not consider the job scheduling based on the reliability aspect and calculate the reliability of the system after scheduling, making them unrealistic. A reliability estimate beforehand does not really reflect the true reliability of the system. Another drawback of the reported models is that they do not focus on all the parameters for reliability estimations. For example some models are topology specific [11, 16], some considers only the hardware failures while some only software failures [19, 4]. Some has even advocated redundancy to increase the reliability [1, 5]. The effect of variation in parameters to reliability was studied in [14] but the study did not propose any allocation strategy while some finds application for only specific grids like database specific girds [2].

In a heterogeneous environment like grid, allocation on the suitable resources is an important activity. Almost all of the models, proposed in the literature [1, 3-5, 11, 13, 15, 16, 18, 19], do not take into account the nature of the job to be allocated. Nature of the job means the specifications and requirements of the job in terms of resource characteristics. For example a graphic application may need a computational resource specialized for graphic specific jobs or a database specific application may demand resources which handles faster storage, data movement and retrieval for efficient execution of the job. Another significant restriction, with the allocation schemes, in these models, is that they do not consider the existing (current) workload on the resources chosen for allocation. Since the jobs preoccupying the resources affect the reliability of the system significantly, the reliability estimates of these models do not represent the true system reliability. Further, the models do not specify how the resource or job attributes like speed of the processing node or communication requirement between the interacting modules affect the resource selection. No clear load balancing strategy has been reported by these models to harness the inherent job parallelism.

3. The Proposed Model

In the proposed model the job is considered to be submitted, for execution, in form of the modules along with the job attributes like number of modules in the job, module size, specialization (nature) of the job, failure rate of each module, module’s precedence in form of its Job Precedence Graph (JPG) and the Inter Module Communication (IMC) requirements which accounts to the amount of interaction between the modules in form of data exchange (in bytes) [18].

The grid comprises of a number of clusters with some specialization. Cluster specialization indicates what the cluster is meant for and is capable of executing a particular type of job much faster as compared to the other jobs. An example could be a grid specializing for the graphic related applications. The model considers a cluster scheduler at each cluster capable of scheduling the job on the cluster nodes. For the allocated modules, each node executes them following its own internal scheduling policy independently. Further, it is assumed that the grid attributes like number of clusters in the grid, number of nodes in each cluster, processing speed of each node, hamming distance between the nodes, failure rate of each node and the links connecting them are available with the scheduler. To reflect the status of the clusters in the grid at any moment of time, a cluster table has been conceived with the cluster scheduler having the following attributes.

\( C_i \) (Node number ‘\( P_k \)’, Processor Clock ‘\( f_k \)’, Specialization ‘\( S_n \)’, Time to finish execution of previous work load ‘\( T_{prev} \)’ for each node, Module identification number ‘\( M_{ij} \)’ for the allocated modules on a node, Node failure rate ‘\( \lambda_n \)’, Link failure rate
Maximizing Reliability with Task Scheduling In a Computational Grid Using GA
Zahid Raza, Deo Prakash Vidyarthi

The Module identification number helps in identifying the association of a module with its job. It is represented as ‘Mi,j’ meaning module ‘i’ of job ‘j’. The cluster tables are always updated periodically to reflect the accommodation of newly allocated jobs.

To apply GA the chromosome structure is designed having number of genes equal to the number of modules in the job. Thus the size of the chromosome is the number of modules in the job with each gene position corresponding to a module and the value at that position indicating the node on which that module is allocated (Figure 1).

Figure 1: Chromosome Structure for a job with ‘n’ modules

Therefore, the allocation pattern for a sample chromosome structure shown in Figure 2, can be interpreted as module 1 allocated on node 5, module 2 on node 6, module 3 on node 5, module 4 on node 5 and module 5 allocated on node 1.

Figure 2: Chromosome structure for a job with 5 modules

This arrangement enables allocation of more than one module to a node. The chromosomes are mated in the successive generations. This results in allocation of modules to different nodes by changing the allocation pattern. The allocation pattern for each chromosome corresponds to some fitness value in terms of reliability of the job execution (as discussed in section 3.2). Since crossover changes the allocation pattern, the fitness value/reliability also changes. The process is repeated by selecting chromosomes with high fitness value and it leads to an allocation pattern with increased reliability over the generations.

3.1 Notation used

The various notation used are explained as follows.

- $N$: Number of clusters forming the grid
- $M$: Number of modules in the job
- $K$: Number of nodes in a cluster
- $C_n$: Cluster identifier
- $S_n$: Specialization of a cluster
- $P_k$: $k^{th}$ node of a cluster
- $f_k$: Clock frequency of node ‘$P_k$’
- $\lambda_{kn}$: Failure rate of the node ‘$k$’ of cluster ‘$n$’
- $J_j$: Submitted job with specialization ‘$j$’
- $m_i$: $i^{th}$ module of the submitted job
- $I_i$: Number of instructions in the module ‘$m_i$’
- $\mu_j$: Failure rate of the module ‘$m_i$’ of job ‘$J_j$’
- $\xi_{kl}$: Failure rate of the link connecting nodes ‘$k$’ and ‘$l$’
- $x_{ijk}$: Assignment vector indicating the allocation of a module ‘$m_i$’ of job ‘$J_j$’ on node ‘$P_k$’. It assumes binary values and is 1 if a module is allocated to a node otherwise it is 0.
- $E_{ijkn}$: Processing time of a module ‘$m_i$’ of job ‘$J_j$’ on node ‘$P_k$’ of cluster ‘$C_n$’. It is less for nodes with higher clock frequency or with fewer loops.
- $B_{ij}$: Inter module communication requirement between a module ‘$m_i$’ and ‘$m_h$’ of job ‘$J_j$’. This becomes zero in the case of non interactive modules or if the modules reside on the same node. This information is assumed to be submitted along with the job.
- $D_{kl}$: Hamming distance between nodes ‘$P_k$’ and ‘$P_l$’ of a cluster. This corresponds to the number of links traversed between nodes ‘$P_k$’ and ‘$P_l$’, if the modules lying on them needs to interact. This information is assumed to be available with the cluster all the time and is updated periodically to reflect the changes.
- $T_{prkn}$: Time to finish work load already existing on a node ‘$P_k$’ of a cluster. This accounts for the time taken by the modules already assigned to the node to finish execution.
- $\text{Rel}_{kn}$: Reliability of execution of module ‘$m_i$’ on node ‘$P_k$’. This is the reliability with which a module can be executed on a node and is a function of the reliability of the module being considered for allocation, reliability of the node on which allocation is being considered and the reliability of the links which participates in data exchange for the interactive modules, residing on different nodes. Thus the same module may have a different reliability of execution on different nodes.
- $\text{Pop}_n$: Number of chromosomes for the cluster
- $g_n$: Current generation of chromosomes for cluster ‘$C_n$’
- $f_g$: Chromosome of generation $g_n$
- $\text{ChromRel}_{kn}$: Reliability offered by the chromosome ‘$f$’ of cluster ‘$C_n$’ with which a job can be executed.
- $\text{ClustRel}_{kn}$: Reliability of execution of a job ‘$J_j$’ at
cluster ‘Cn’. 

- **GridRelJ**: Reliability offered by the grid to the job.
- **SendAll**: Keyword indicating the communication of the job to all the clusters in the grid
- **ReceiveAll**: Keyword indicating communication of the ‘ClusRelJn’ from all the selected clusters to the calling node

### 3.2 Fitness function

To realize a realistic grid picture it is essential to account for the contribution of the various constituents towards job execution viz. the resources on which the job is being considered for execution, job characteristics and the environment in which the job will execute. The contribution of the node attributes can further be classified in terms of its processing speed ‘Eijkn’, which reflects the node’s computational capabilities, its failure rate ‘λik’ and the existing workload on the nodes ‘Tprkn’. The computational capabilities of a node derived as ‘Eijkn’ is the processing of a module ‘mi’ of job ‘Jj’ on node ‘Pk’ of cluster ‘Cn’. This is obtained while considering the job allocation in a cluster as

\[
E_{ijkn} = \frac{(Number \ of \ instructions \ in \ a \ module(Ii))}{Clock \ speed \ of \ the \ node(fk)} + Loop \ overhead
\]

where ‘l’ is the multiplication factor to account for the loop constructs in the module and ‘α’ is the average time taken in the loop.

Job characteristics also affects the job execution with respect to the failure rate of its various modules ‘µij’, and degree of interaction between the modules. The grid environment affects the same as the data is transferred on the links connecting various nodes assigned with interactive modules and these links are also vulnerable to failures with a failure rate ‘ξkl’. Therefore, the reliable job execution depends on

1. Reliability of the application (job), which is submitted in the form of modules.
2. Reliability of the nodes on which the job is being considered for allocation and
3. Reliability of the links through which modules residing on different nodes may communicate.

Further, the computing nodes are required to be reliably available for the time the module is being executed on that node. The links should also be reliable during the time the inter module communication takes place. The module failure rate is determined at the time of the application development and is taken care of by the software engineering paradigm.

Considering all these factors, the reliability of execution of a module ‘mi’ on a node ‘Pk’ can be stated as

\[
ModRel_{ik} = \exp \left\{ - \left[ (\mu_{ij} - \lambda_{ik})E_{ijkn,x_{ik}} + (\mu_{ij} + \xi_{kl}) \sum_{h=1}^{M} (w(B_{ij},D_{ih}))x_{ikh,x_{ih}} + \lambda_{ik}T_{prkn} \right] \right\}
\]

where \[ \sum_{h=1}^{M} (B_{ij},D_{ih})x_{ikh,x_{ih}} \] is the cost of communication corresponding to the interactive modules and ‘w’ is the scaling factor to scale this cost into time unit. For every module allocated on a node, the time to finish pre-assigned workload ‘Tprkn’ gets modified as

\[
T_{prkn} = T_{prkn} + \sum_{h=1}^{M} (B_{ij},D_{ih})x_{ikh,x_{ih}}
\]

Each chromosome has an allocation pattern referring to scheduling the nodes to the modules. Therefore, each node on which allocation has been made corresponds to a reliability ‘ModRel_{ik}’, it offers to the modules allocated to it. In a cluster, for a chromosome, considering ‘ModRel_{ik}’, for all these nodes results in a reliability for the whole job consisting of ‘M’ modules referred to as the Chromosome reliability ‘ChromRel_{jn}’ and is calculated for all the nodes on which the allocation has been made as

\[
ChromRel_{jn} = \prod_{i=1}^{k} ModRelCost_{ik}
\]

where ‘k’ is the number of modules.

All the chromosomes in a generation correspond to a reliability it offers to the job. Thus the chromosome which offers the highest reliability is treated as the fittest chromosome with its reliability being the reliability offered to the job by the cluster ‘ClusRel_{jn}’ calculated as

\[
ClusRel_{jn} = \max (ChromRel_{jn}) \quad (for \ f=1 \ to \ PopSize)
\]

Similarly, ‘ClusRel_{jn}’ can be obtained from all the clusters and the job can be allocated to that cluster offering the highest value of ‘ClusRel_{jn}’ referred to as the Grid Reliability ‘GridRelJ’ as

\[
GridRelJ = \max (ClusRel_{jn}) \quad (for \ all \ selected \ clusters)
\]

4. **Working of the Model**

The job is submitted as specified in Section 3. The
Maximizing Reliability with Task Scheduling In a Computational Grid Using GA
Zahid Raza, Deo Prakash Vidyarthi

job can be submitted at any node of any cluster in the grid in a multipoint entry system. The algorithm first searches for the suitable clusters by checking specialization of the cluster reflected in the cluster table. If it matches with the nature of the job, the clusters are further examined for estimation of the reliability it offers to the job.

For each selected cluster, a population of size ‘PopSize’ is generated. Each gene of the chromosome is randomly allocated a node. This reflects module assignment to a node. The reliability ‘ModRelik’ with which these modules can be executed on the respective nodes is then calculated as per equation (ii). After evaluating the ‘ModRelik’ for the individual nodes, as per the allocation pattern suggested by the chromosome, the reliability offered by the chromosome ‘ChromRelin’ is estimated as per equation (iv). For the current generation the chromosome ‘fgn’ offering the highest reliability is then marked as the fittest chromosome. This chromosome is mated with the remaining chromosomes on a given crossover point using single point crossover to generate a population of twice the size of the original population. The population is then again restored to its original size by the selection of those chromosomes with best ‘ChromRelin’. This process is repeated over a number of generations to result in evolution and selection of a chromosome ‘fgn’ offering the best reliability corresponding to its allocation pattern. The reliability corresponding to this chromosome can be treated as the best reliability offered to the job till that generation. This reliability eventually becomes the reliability offered to the job by the cluster ‘ClusReljn’ after last generation. Similarly, ‘ClusReljn’ from all the selected clusters are compared to select the cluster with the highest reliability known as the ‘GridRelj’. Fifth generation mutation is effectuated in our experiment to bring the result out of the local maxima. This is done by randomly picking a gene from the chromosomes and replacing the node on that gene by some another node generated dynamically.

```
Alloc (Job)
{
  Submit the job in form of modules
    // Submit the job in form of modules mi (i = 1 to M)
    // in the desired format
  Sendall
    // Send the jobs to all the clusters of the grid for
    // evaluation of their suitability for the job execution
  For all clusters Cn (n = 1 to N), do
    // Match the specialization of the job with the cluster
    
Select the clusters with matching specialization
For the selected clusters, do
{
  Calculate reliabilities ( )
}
// Estimate the reliability offered by the clusters to the job
Compute Pjk
  // Calculate the processing time of each node,
  // for all the modules, as per eq. (i)
Generate population
  // Generate chromosomes of size ‘PopSize’ with
  // number of genes equal to the number of modules.
  // Randomly allocate nodes to these modules.
For each generation, do
{
  For all chromosomes, do
    Compute ModRelik
      // Calculate the reliability of job execution on each
      // node Pk (k=1 to K) on which allocation has been proposed
      // as per eq. (ii)
    Compute ChromRelin
      // Calculate the reliability offered to the job by each
      // chromosome as per eq. (iv)
  Perform Selection
    // Select the chromosome fgn with highest ChromRelin
  Perform Crossover
    // The selected chromosome fgn is mated with
    // the remaining chromosomes for the specified
    // crossover point.
  Perform Mutation in Fifth Generation
    // For every fifth generation randomly replace any
    // gene value in the chromosomes with a new value
    // to avoid local maxima
  Compute ModRelik
  Compute ChromRelin
  Restore Population
    // Select half of the population with highest ChromRelin
  
Receive all
Compute GridRelj & select the cluster for job execution
  // Calculate GridRelj as per eq. (vi)
  Send the job to the selected cluster
}
```

```
This essentially alters the node allocation pattern for the chromosome and hence the reliability offered to the job by the chromosome. The algorithm for the process is as above.

5. Simulation Study

Experimental study is done to observe the behavior of the model under varying grid conditions. The experiment assumes job submission at any node of the cluster. The reliability for executing this job is then estimated on the clusters whose specialization matched with the job. All the data values conform to the other models for the similar study and are generated dynamically. The failure rates $\lambda_{ij}$, $\mu_{ijk}$ and $\zeta_{kl}$ randomly vary from 0.0000 to 0.0009. Table 1 illustrates some of the important results for some of these experiments.

<table>
<thead>
<tr>
<th>Job No</th>
<th>No. Of Modules in the Job</th>
<th>Reliability offered by the selected clusters</th>
<th>Reliability offered by the grid</th>
<th>Allocation Pattern for the finally selected cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job $J_0$</td>
<td>8</td>
<td>0.6921, 0.6539, 0.5981</td>
<td>0.6921</td>
<td>5,5,2,4,3,6,9,3</td>
</tr>
<tr>
<td>Job $J_0$</td>
<td>12</td>
<td>0.3983, 0.4701</td>
<td>0.4701</td>
<td>1,2,2,5,1,6,5,6,7,7,2,2</td>
</tr>
<tr>
<td>Job $J_0$</td>
<td>14</td>
<td>0.2855, 0.2700, 0.2911</td>
<td>0.2911</td>
<td>7,7,7,5,4,4,3,3,7,1,6,10,10</td>
</tr>
</tbody>
</table>

The noteworthy observations from the experiments conducted are as follows

- Clusters are selected on the basis of their matching specialization with the job.
- Allocation depends on the failure rate of the nodes, network links and the application.
- The allocation takes into account both the processing speed ($E_{ij}$) of the node and the communication requirement of the modules being considered for allocation with the previously allocated nodes i.e. $\sum (B*D)$ for reliability estimations. Further, the effect of the previous workload is also accounted for reliability estimates depicting a true picture of the grid.
- The job is scheduled on the node which offers the maximum reliability at that moment of time in the grid.
- The allocation takes care of load balancing properly as the modules are distributed evenly on the suitable nodes.

- More number of nodes may not offer better reliability to a job as the node (cluster) selected is the best tradeoff between operating speed, failure rates of various parameters and the workload assigned to the nodes of the cluster.

Since GA has been used in this model a simulation study was conducted to study the variation in the reliability over the generations. The experiments were conducted till fifty to seventy generations for a single point crossover. Figure 3 represents the results of this study for some experiments.

It is observed form Figure 3 that the reliability increases over the generations and tends to saturate gradually thereby improving the reliability of execution of the job submitted. However, the rate of convergence is observed to depend on the crossover site. It is normally observed that a cross over point chosen towards the higher side of the chromosome led to faster convergence of the results. Further, it is noticed that over the generations the model tends to favor those chromosomes offering a uniform distribution of modules over the nodes as compared to the ones with non uniform distribution. This could be explained as increasing modules on a node reduce the reliability whereas if the load is properly distributed over the nodes it results in a more reliable grid picture. This selection eventually becomes responsible for the improvement in the reliability over the generations. The experiments were conducted for many different set of data and a similar trend was noticed in all the experiments.

5.1 Comparison with non GA model

The present work focuses on scheduling the job to those resources which offers the maximum reliability to the job execution. Earlier a model was proposed by the authors for the same objective without using genetic algorithm being referred as the non GA model [14]. Simulation experiments are conducted to compare the performance of the proposed model with the non GA model. The non GA model works on the basis of
evaluating the reliability offered by all nodes of the cluster for each module sequentially as per its Job Precedence and Dependence Graph (JPDG) which provides the order of execution of the modules in the job and their interaction requirements (if any). The module is allocated to that node which offers the maximum reliability. This process is repeated for the remaining modules of the job. The resultant allocation reflects the allocation proposed by the cluster. 'ChromRel\textsubscript{inj}' and 'GridRel\textsubscript{j}' are then evaluated in the same manner as used by the proposed model using equation (iv) and (v).

Performance comparison of the proposed model with the non GA based model is done by providing the same grid environment to both the models and the results, in terms of the reliability offered to the job are compared for a suitable crossover point. It is found that over the number of generations the result of the GA based model converges towards the one offered by the non GA based model and many times even offered a better solution depending on the generations and the cross over point used. For the single point crossover, it is observed that the difference in the reliability offered by the two models varied between -2% to +2%. Figure 4-5 represents the comparison of the performance of GA based and non GA based models for two such experiments reflecting the same results with the observation that the reliability offered by the model gradually moves closer to the one offered by the non GA model.

Further, figure 6 represents a general comparison of the execution rates obtained from various experiments for the GA based and non GA model. Thus GA based model, successfully scheduling the job on the reliable resources, can be rated equally good as compared with it's non GA counterpart.

6. Conclusion

Reliability of job execution is an important issue for a dynamic system like grid from system design and evaluation point of view. As the resources in the heterogeneous grid environment may fail at any time, scheduling a job on the reliable resources becomes very important for the successful execution of the job. Most of the work reported in the literature does not take into account the dynamic nature of the grid. Further, none of the models schedule the job based on the reliability offered by the grid. The reported models consider the reliability estimates based on one or the other parameter of the grid and assuming the scheduling of the job already done. Therefore, these models do not stress on the scheduling strategy that need to be adopted. Further the earlier work reported do not, in any way; consider the effect of the existing workload on the reliability of the grid.

The proposed model considers the reliability of the computational resources, the network links along with the application reliability. Accordingly, the job is scheduled to those resources which offer the maximum reliability at any point of time. It is different from the works reported in the literature in the sense that it presents a realistic picture of the scheduling on the grid by estimating the reliability of the grid from both hardware and software point of view and then allocating the job based on the reliability estimates. Therefore, rather than estimating the reliability of the grid assuming the job already allocated, the model actually allocates the job based on reliability. This approach is more suitable to the dynamic nature of the grid. Since the reliability of the grid is affected by the workload already
present on the nodes, the model takes care of this situation also for the reliability estimates. This fact is observed in the experimental results also with the results favoring the chromosomes with uniform load distribution. Further, allocation of the modules based on the specialization of the job makes resource selection and allocation more specific to the job requirements eventually leading to the most appropriate resources being selected.

A comparison is made for the present model with a non GA based model developed by the authors using the same allocation strategy. It is found that the GA based model effectively schedules the job and many times even offers a better solution.

Ensuring reliability based scheduling can help in developing and deploying complex scientific applications on the grid which requires reliable computation. Further, the approach can open up and strengthen the much awaited commercialized use of the grid leading to ubiquitous computing.

7. References