Journal of Theoretical and Applied Information Technology
10th December 2013. Vol. 58 No.1

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

A NOVEL DYNAMIC RELIABILITY OPTIMIZED RESOURCE SCHEDULING ALGORITHM WITH FAULT TOLERANT APPROACH FOR GRID COMPUTING SYSTEM

[1]U. SYED ABUDHAGIR, [2]Dr.S.SHANMUGAVEL

[1,2]Department of Electronics and Communication Engineering, College of Engineering, Guindy, Anna University, Chennai-600025.

E-mail: [1]syedabudhagir@gmail.com, [2]ssvel@annauniv.edu

ABSTRACT

In this paper, we design global optimization model and fault tolerance service for grid computing system. It is provided as a promising model for grid resource scheduling algorithm. It aims at solving the problem of optimally allocating services on the grid to optimize the grid service reliability, deadline and cost. In this paper, the problem of optimizing the reliability of grid systems has been modeled as a multi-objective optimization problem where apart from the grid system reliability; the system cost and redundancy are also considered as objectives. For the reliability analysis of the grid system, we consider failure rate of computational resources and network resources. Based on the service reliability of the grid system, Our IGA-RORS algorithm selects the set of optimal resources among the candidate resources based on reliability, application execution time and cost that achieves optimal performance using an immune genetic algorithm. In our algorithm, we design and implement a fault tolerance service; it guarantees the completion of the applications in the optimally selected resources. We have demonstrated the test results in java based GridSim tool.

Keywords: *Grid Computing System, Resource Management System, Reliability, Redundancy, IGA, Fault Tolerant service.*

1. INTRODUCTION

A grid computing system is the application of several computers to a scientific or technical program that requires a large number of computational as well as storage resources for processing and accessing large amounts of data [1, 2]. The grid system contains a series of computational, network and storage resources.

The resource sharing and (job) application scheduling is controlled by a resource management system (RMS) [3]. Once the RMS gets a service demand from a grid user, it divides the applications into a several tasks that are executed using the requested resources. After the execution of the tasks, the computed results are return back to the RMS which combines the obtained results into the whole process outcome that has been asked by the customer. The execution time of any grid applications is a random phenomena affected by several factors [4, 5, 8]. There are many computational resources which are available online with different task processing capability.

Thus, the job completion time can vary depending on which resource is assigned to execute the task. Some network and computational resources can fail while assigning and running the job and therefore, the execution time is also affected by the grid system reliability. Some of the earlier works have considered the reliability of the resources parameter while evaluating the performance [5, 6]. In practical, the performance and reliability are related and impact each other in particular, when grid computing is implemented.

When an application is completely parallelized into n different tasks implemented by n resources at the same time, the performance can be improved when the reliability of the network and computing resources are high. Otherwise, it will cause the application to restart, demanding a number of redundant computational and network resources to perform the same tasks, particularly for those failure-prone grid resources. Assigning many redundant resources for the applications may improve the system reliability can reduce the performance of the system by not fully utilizing the resources with more jobs assigned to it [7, 10]. Thus, the performance, and the reliability should be examined together.

Considering the large number of different fault tolerance techniques (FTTs), selecting the best available technique under different levels of faulty system with different system's performance requirements is not an easy task. The system centric

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

approach of FTT is more efficient because it enhanced the system throughput, utilization, complete execution time and economic profits [22, 24].

This paper is structured as; Section 2 provides proposed Optimized Grid Resource Management System (O-GRMS). Section 3 presents grid service model, reliability model and. Section 4 presents problem formulation and reliability optimization model. Immune Genetic Algorithm based Reliabiltiy Optimized Resource Scheduling Algorithm's performance evaluation is presented in section 5. Section 6 offers some conclusions and directions for future directions of this work.

2. PROPOSED O-GRMS ARCHITECTURE

Whenever the Grid Resource Managing System (GRMS) accepts an application, it contacts a Grid information service like the Grid Resource Directory (GRD), to query about available services for each task and their QoS attributes. Each Grid Resource node has to register itself with the GRD, so that it can offer its services to users. Using this information, the O-GRMS can execute a scheduling algorithm to map each task of an application to one of the available services. Then, the GRMS directly contacts the resources to query about the free time slots of the suitable services. Using this information, the GRMS can execute a scheduling algorithm to map each task of an application to a set of the available resources.

Fig.1 depicts our proposed Optimized-Grid Resource Management system model, which includes application queue manager, optimizer, scheduler and fault manager in a grid computing environment. This proposed model is similar to the one with modification from the model described in [3, 24]. It is assumed that all parallel tasks, along with information provided by applications, are submitted to the O-GRMS. An Application Queue Manager (AQM) for incoming applications is maintained by the O-GRMS, which in turn sends the application to Grid Resource Discovery (GRD) module. We assume that the application is partitioned into several tasks and some of the task may have data dependency.

Implementation of a distributed scheduling algorithm is more difficult and complicated than centralized scheduling model.

In distributed scheduler model, the scheduler is dedicated to scheduling the applications and the computing power tends to be underutilized, especially when the schedulers are idle. On the other hand, if the schedulers are able to serve as processing elements when they have no job to schedule, it is difficult to predict when the schedulers will be idle in a dynamic cluster environment [15]. Therefore, the centralized optimizer and scheduler are employed in our O-GRMS model.

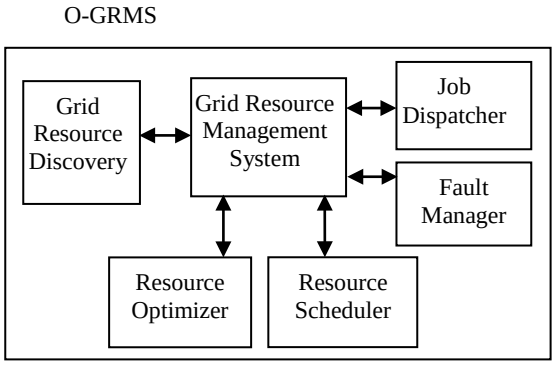

Figure 1: Proposed O-GRMS Architecture with IG-RORS Scheduling Algorithm

2.1. Grid Scheduling and Reliability Model

A grid application is modeled by a Directed Acyclic Graph (DAG) G= {V, E}, where V = {v_1, v_2,., v_n} represents a set of tasks, and E is a set of arcs. Each arc e_{gh} = (v_g, v_h) represents a data precedence constraint that indicates that task v_g should complete its execution before task v_h can start. In a task graph entry task means a task without parent task, exit task refers to a task without child task.

A linked set of tasks form an application. The effective realization of an application is obtained if all the tasks have independently been carried out efficiently. The effective process performance is dependent on the application and its specifications.

Here, we consider a set of applications A = $\{A_L\}_{L=1,\ldots.k}$ with n tasks can be described,

$$A_L = \sum_{i=1}^{n} T_i \quad (1)$$

where T_i, are the tasks. A heterogeneous grid is modeled by a set R_j = {R_1, R_2, . . . , R_m} of resources, where R_j is a computational resource with local memory.

In our model, a service is defined as a computational resource and the required software to execute an application (job).

There are many QoS attributes for services such as deadline (job completion time), cost, reliability and security. In this study, we have taken reliability, deadline and cost for scheduling the application. The cost and execution time of a

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

service usually depends on reliability of the resources and links.

2.2.1 Service Reliability model

The reliability model which is similar as the one defined in [6], assumes that temporary failures occur according to a Poisson probability distribution and failures are mutually independent.

Let Y be an $m \times n$ binary matrix corresponding to a schedule, in which m tasks of a job are assigned to n processors. The element y_{ij} equals 1 if and only if v_i has been assigned to R_j; otherwise $y_{ij} = 0$. In reliability model proposed in [15], the failures of resources, nodes and links during idle time are not considered, because during this period failures can be fixed. Reliability is calculated based on the packet loss rate and bit error rate is presented in [21]. In our work, the failures are taken in account to dynamically evaluate the performance of grid service reliability in a more realistic and optimized way. The quantity of data that transmitted among the O-GRMS and resource R_j that processes task v_i is denoted by d_i. If the data transmitted between the O-GRMS and resource R_j is accomplished through links belonging to a set φ_j, then the data transmission speed is

$$s_{i,j} = \min_{x \in \gamma_j}(bw_x) \quad (2)$$

where bw_x, is the bandwidth of the network link L_x between O-GRMS and resource R_j. Communication time of task v_i from GRMS to resource R_j can be denoted by,

$$CT_{i,j} = \frac{d_i}{s_{i,j}} \quad (3)$$

The estimated execution time ($EET_{i,j}$) of task v_i executed by resource R_j can be written as,

$$EET_{i,j} = RCT_{i,j} + CT_{i,j} \quad (4)$$

Where $RCT_{i,j}$, resource computation time of task v_i on resource R_j.

Based on the conventional models [13], we assume the failures occurring on the resources, nodes and links satisfying Poisson processes and the failures on different elements are independent from one another, which is a good approximation since most shared resources are located a far distance from others in large-scale grid computing systems. We assume application A_L can be executed at any resource-managing sites.

Table 1: Notations

GA – Genetic Algorithm
IGA – Immune GA
SP – Selection Population
PC – Crossover Population
PZ – Population Size
IS - Immune Selection
VP - Vaccination Population
G-RORS – GA based Reliability Optimized Resource Scheduling
IG-RORS – Immune GA based Reliability Optimized Resource Scheduling
$R_G(\gamma)$ – Reliability of the grid system
$R[M_q(x)]$ – Reliability of the MRST
R_d - Redundancy level
B - Budget allocated by the user for the application
TC_j - Total execution cost of the application in grid system
NC_j - Communication cost between resource j and O-GRMS for task i
RC_j – Computational cost of the resource j to execute the task i
AD_i – Deadline for i^{th} Application
$TE_{i,j}$ - Total application execution time
$CT_{i,j}$ – Communication Time of task v_i from O-GRMS to resource j.
$EET_{i,j}$ - Estimated Execution Time of task v_i executed by resource Rj
OT – Optimization Time
ST – Scheduling Time
λ_{OG} - Failure rate of O-GRMS

Let Mq(x) denote the q^{th} minimal resource spanning tree (MRST) induced from a resource schedule design x [9, 10] and let $R[M_q(x)]$ denote the service reliability of Mq(x). The MRST is defined as a resource spanning tree RST_j such that there is no existence other task spanning tree RST_j which is subset of this RST_j. The service reliability of the MRST, ($R[M_q(x)]$) can be simply expressed as,

$$R[Mq(X)] = e^{-\lambda_{OG} * TE_j * \beta_j} * \prod_{NL_{i,j} \in S_l} e^{-\lambda_{i,j} * (TCT_{i,j})} * \prod_{j \in S_l(z)} e^{-\lambda_j * (RC_j)} \quad (5)$$

where $\lambda_{i,j}$ denotes the failure rate of the link L(i, j); L(i,j) denotes the link between the node N_i and node N_j; λ_j denotes the failure rate of the node N_j. β_j denotes packet loss rate between O-GRMS to resource R_j.

Assuming there are Q, MRSTs for a given resource allocation design x, then the service reliability can be computed by $R[M_1(x) \cup M_2(x) \cup \ldots \cup M_q(x)]$, which represents that the system works if at least one of the MRSTs works.

Journal of Theoretical and Applied Information Technology
10th December 2013. Vol. 58 No.1

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

3. RESOURCE OPTIMIZER MODEL

We consider multi objective reliability optimization problems which have one objective function, such reliability optimization problem of redundant resource, cost and application deadline as its constraint. In many of the reliability and performance based scheduling approach [17, 18,19, 20], they have not considered the optimization time and scheduling time while achieving the deadline. In our model, both these two parameters have been taken into consideration while modeling the system. We assume the existing grid computing system consists of N grid nodes denoted as GN_i and possesses M computing resources denoted as R_j. The original computing resources are selected on the accessible nodes with a maximum grid service reliability $R_G(\gamma)$ using the minimum budget and deadline. The resource scheduling of computing resources problem is formulated as follows.

3.1 Optimization Model

Using redundant resources as techniques is well accepted to improve the reliability of system [14]. This problem can be normally formulated as nIP model. The $\gamma_{i,j}$ represents the allocation of the jth resource (R_j) on the i[th] node (G_i): $\gamma_{i,j}$ = 0 means that the R_j is not integrated on the G_i and $\gamma_{i,j}$ = 1 means that R_j is integrated on the G_i. γ is defined as a vector of $\{\gamma_{i,j} | \ i \in [1,N], \ j \in [1,M]\}$, which represents an allocation schedule of the grid service resources on the nodes. Hence, given the structure of the nodes and links involved in the service, the grid service reliability can be determined by the allocation-scheduling vector γ. The grid service reliability of the MRST can be computed by Eq. (5). Thus, the optimization problem becomes to find the optimal solution of s so that the integrated grid service reliability is maximized. The optimization model is given below:

Decision variables:

$$\gamma = \{\gamma_{i,j} = 0,1 | \ i \epsilon [1,N], j \epsilon [1,M] \} \quad (6)$$

Objective function:

$$\text{Max } R_G(\gamma) \quad (7)$$

Subject to:

$$\sum_{i=1}^{N} \gamma_{ij} \geq 1 \ \ j = 1,2 \ldots \ldots \ldots M \quad (8)$$

$$\sum_{i=1}^{N} \gamma_{ij} \leq R_{d,j} \ \ j = 1,2 \ldots \ldots \ldots M \quad (9)$$

$$\sum_{j=1}^{M} \gamma_{ij} \leq \eta_i \ \ i = 1,2 \ldots \ldots \ldots N \quad (10)$$

$$\sum_{j=1}^{M} TC_j \sum_{i=1}^{N} \gamma_{ij} \leq B \quad (11)$$

$$\text{Where, } TC_j = \sum_{j=1}^{M} [Rc_j + Nc_j] \quad (12)$$

$$\sum_{j=1}^{M} TE_j \sum_{i=1}^{N} \gamma_{ij} \leq (AD - OT) \quad (13)$$

Where,

$$TE_j = \sum_{j=1}^{M} [EET_j + CT_j + ST_j] \quad (14)$$

where $R_G(\gamma)$ denotes the grid service reliability of a resource scheduling design $Y^{(0)}$; $Y^{(0)} = [Y_{i,j}^{(0)}]$, i = 1,…N, j =1,...M, denotes a constant matrix representing the originally scheduling of computing resources such that $Y_{i,j}^{(0)}$= 1 implies the computing resource R_j is allocated in node GN_i and $Y_{i,j}^{(0)}$ = 0 otherwise; The first constraint (8), represents that atleast one resource in the GN_i is to be selected. The second constraint (9), η redundancy of each job should not exceed the maximum allowable level. The third constraint (10), that the number of computing resources R_j in node GN_i should not exceed α_i, its upper bound is based on the size of job in application. The fourth constraint (11) user defined budget for the application, scheduling a job to the resource should not exceed prescribed value. The fifth constraint (13), user defined application deadline, here approximate scheduling and optimization time is taken from the GIS database, scheduler should submit the job to resource and it finishes the computation within the prescribed deadline.

4. PROPOSED SCHEDULING ALGORITHM (IGA-RORS)

Algorithm 1 shows the pseudo code of the overall IGA-RORS algorithm for scheduling grid applications. After collection of information of grid resources, network link and grid nodes characteristics the suitable resources have been discovered for executing the applications. In lines (5-7), estimate reliable resources set for the application, optimizing reliability of the application execution based on the primary constraints which are completion time (deadline) and budget, and secondary constraint is redundancy by using immune genetic algorithm and resource set information send to the fault manager. In line 8, we used checkpointing method in fault manager, if any of the resources fails during the execution period, our algorithm calls fault manager finds reliable resource and send the last saved checkpoint value.

Algorithm 1: IGA based Reliability Optimized Resource Scheduling Algorithm (IGA-RORS) with Fault Tolerant

Input : Application set A = {A_1, A_2 . . . A_T}, Grid node and resource set

Output : The application and Grid node schedule pair < A_T, GN_N, R_K >

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

1: Collect the information of the resources which are connected with the RMS such as computing and network resource property which include cost, MIPS rating, link bandwidth, memory, failure and packet loss rate.
2: While there are applications to schedule do
3: for each Application A_i do
4: Identify, which resources can execute the user's application through the GIS.
5: Estimate the reliability of the MRST by using the equation (5).
6: Call Relioptim- IGA algorithm, which returns the best MRST set for A_i.
7: Assign the A_i to best Reliable optimized resource ROR_i set and send assigned resources set information to the fault manager.
8: Fault manager Calls FTT algorithm.
9: end for A_i
10: If (size (A_{i+1}) <=size (A_i) && AD_{i+1}<< AD_i.
Assign A_{i+1} to the ROR_i set.
11: else repeat the step 4
12: end while
13: end procedure

4.1 Estimate Reliability Resource Algorithm

In this algorithm, based on the equation (7) minimal resource spanning tree is calculated for the given application. The pseudo code for Estimate Reliability Resource is shown in Algorithm 2.

We assume task v_i can be executed at any resources. The O-GRMS is denoted as a root node GN_v in the following deviation. The reliability of the task running on resource R_j, $R[M_q(\lambda)]$ can be simply expressed as in (5).

Algorithm 2: Estimate the service reliability of Grid using ER Resource algorithm,

1: Calculate the data of all the MRSTs for a given task-resource list y, i.e. $MRST_1(y)$, . $MRST_L(y)$.
2: Calculate reliability of each MRST by using equation (5) and order them.
3: Take M1(x) as the first MRST.
4: We proceed with the above procedures until
$R_q[MRST_1(y) \cup MRST_2(y) \cup..\cup MRST_L(y)] - R_q[MRST_1(y) \cup MRST_2(y) \cup..\cup MRST_{L-1}(y)] \leq \varphi$
where φ is a predetermined small positive value.

The resulted grid service reliability will be served as,
$R_q[MRST_1(y) \cup MRST_2(y) \cup..\cup MRST_L(y)]$.

After generating all the MRST's, any one of the MRST's to be operational can guarantee the grid application to be executed successfully.

4.2 RELI-OPTI IGA

We analyzed above presented reliability optimization model with GA.

The fitness function used in this model is given by,

$$Fitness\ F = \frac{-A}{\ln(R_G(y))} \quad (15)$$

where, A is constant.

In GA, the main genetic operators are crossover and mutation, not only give each individual's the evolutionary chance to attain global optimum but it lead to the degeneracy to some degree because of random and unsupervised searching during the entire process. On the other hand, GA is lack of capability of making use of some basic and obvious characteristic or knowledge in pending problem. Based on the above considerations, immune Genetic Algorithm [23] is proposed.

Algorithm 3 shows the structure of immune genetic algorithm. The solution after crossover is taken for immune operations. IGA is a one of the intelligent optimization algorithm, which mainly constructs an immune operator accomplished by two steps: Immune selection and Vaccination. The initial populations are depends on the number of jobs in grid application. Random selection method for parent selection and 10% of best solution is taken to next generation. The knowledge added IGA algorithm performed in the following way.

Algorithm 3: Reli-Opti Immune Genetic Algorithm

IGA (G = (V,E), SP, CP, PZ, MG(maxgen))
Step 1: for loop=1:PZ
Pop = initial population using jobs number
end for
Step 2: while (until reaches optimal solution is identified)
PSP = selection (PS, population);
//uniform crossover is done on the selected parents
Step 3: CP = crossover (CP, SP);
Step 4: Immunization (CP)
// solutions that satisfies primary and secondary constraints are taken by immune selection
IS = ImmuneSelection (CP);
//resulting solutions are arranged according to reliability of the MRST set. Vaccination is

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

performed based on resource allocation matrix
VP = Vaccination (IS);
Step 5: Evaluate the fitness of each VP
Step 6: If optimal solution is identified then
break;
else
// replace the best few offspring with the initial population and continue the loop
Survival (initial population, VP);
end if
end while

4.3 Fault Tolerance Technique:

In our fault manager, we used Check-pointing technique for fault tolerance [22] is the process of saving the current state of the tasks/ applications, so that whenever a network or resource fails during the execution time, it migrate the particular task into a another faster and highly reliable available resource in the resource set pool and starts the execution from the last saved checkpoint. This reduces the execution time, cost and increases reliability of the grid system.

Step 1: Revised Execution time of the last saved checkpoint is calculated by using the equation (2), (3) & (4), Revised Execution Time (RET) of the last saved checkpoint of the task,

$$RET(A_i(v_i)) = CT(v_{i,j}) + EET(R_{i,j}) \quad (16)$$

Step 2:
If, $REC(A_i(v_i)) < AD_i$, then assign the task to the newly discovered resource among the highly reliable resource.
Else repeat step 2.
Step 3: End Procedure.

5. SIMULATION AND DISCUSSIONS:

In order to verify the efficiency of the proposed IGA-RORS algorithm, we have carried out simulation studies in java based GridSim tool. Considering the NP-completeness features of the resource scheduling issues for data precedence constrained jobs, it is challenging to create analytical analysis of the above proposed algorithm.

The main constraint is that experiments on real world grid platforms are often non- reproducible due to dynamic availability of the grid resources. In this environment comparing different scheduling algorithms are very difficult. For this reason, simulation is the best choice in order to test and compare scheduling algorithms (14). We use the GridSim simulator, it supports modeling and simulation of heterogeneous grid resources, users and application models. It provides framework for establish a grid like environment with creation of applications and their management.

First we generated grid system, it consisting of 10-100 grid nodes. Each node attached with minimum of 10 resources to maximum of 100 resources. The bandwidths of all links are uniformly distributed between 50Mbps and 1 Gbps. Processor capacity varies from 500 to 10000 MIPS, each node's computing delay varies from 1 to 20 per time unit. To allow grid task agents to complete tasks, an6 additional margin of 300 time units is provided.

The initial value of the application budget denoted by B varies from 10 to 500. Completion times and resource allocation efficiency are two measurement criteria to measure in the experiment. The failure rates of network links, processors and packet loss rate are assumed to be uniformly distributed among 0.00005 to 0.0000092, 0.00002 to 0.00009 and 0.000002, 0.0000009 respectively.

As Grid is geographically distributed and dynamic system, the failure rate of Grid resources, network links are changing real time. This system information can be updated in GIS, which provide information to the scheduler, at a fixed interval. In order to verify the efficiency of the proposed reliability optimized resource scheduling algorithm, simulation studies were carried out to evaluate the performance of IGA-RORS algorithm with the GA based RORS without FT, GA-RORS with FT and IGA-RORS without FT. We compared with our proposed algorithm in terms of application success rate and scheduling efficiency.

The first experiment is to measure scheduling efficiency under different grid size. From Figure 2, the simulation result shows that the scheduling efficiency of G-RORS without FT, with FT and IG-RORS without FT are nearly same when the grid size is 50, but IG-RORS gives better result even small grid size. We simulated Grid size up to 1000, IGA-RORS with FT outperforms the remaining algorithm presented here by 20.4%, 9.5% and 11.4%.

In our second experiment, we measure the success rate of the application by varying the number application. The simulation result in figure 3 shows that the success rate of completion of the submitted applications to the grid is 99.95% by using our approach IG-RORS with fault tolerant and 99.5% by without FT.

And based on GA approach, we got 99.6% with FT and 99.2% without FT when the lesser

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

number of applications is given to the grid. When we submitted higher number applications, we have taken 100 applications, our proposed algorithm gives success rate of 99.6% which better than G-RORS with FT 94.6%.

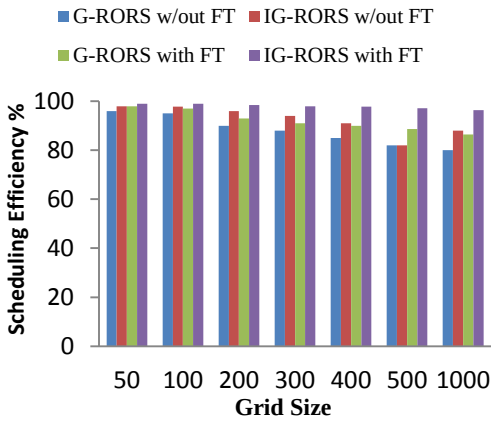

Figure 2: Grid size vs Scheduling Efficiency

We can improve performance and success rate of the grid computing system by 5.28%. In the case of without FT also IG-RORS outperforms G-RORS with the 7.1%.

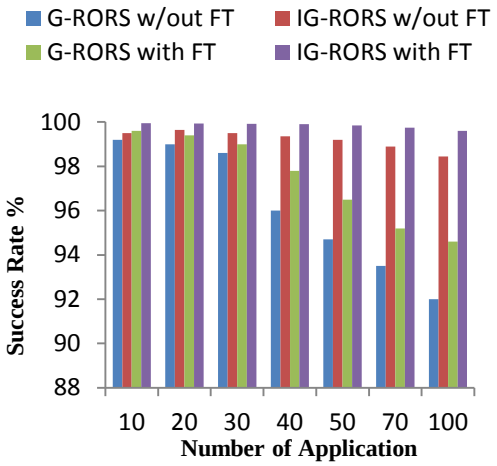

Figure 3: Number of Application vs Success Rate

Through simulation, we confirmed that our proposed algorithm, selects the set of optimal resources and the fault manager monitors the network and computational resources about the state of resources and detects the happened failures, it guarantees execution of the submitted jobs and improves the performance of job execution.

6. CONCLUSION

In this work, we presented an Immune GA based reliability optimized scheduling algorithm named IGA-RORS for grid application scheduling that successfully execute the application while achieving minimizing the deadline and budget and maximizing the reliability. It has three phases: estimate grid service reliability, Reliability Optimization and Fault Tolerant. In the estimation grid reliability, network and computational sensitive resources are discovered, its reliability is calculated based on the failure rate of network and computing resources. Immune GA is used to optimizing the resource and a checkpointing based fault tolerant service to guarantee that the submitted grid applications would be completed reliably and efficiently. We implemented a fault tolerance in grids and performed the simulation to measure the performance improvement due to optimal resource selection and resources fails during execution. We tested our approaches through extensive simulations and obtained promising experimental results.

We evaluate our algorithm by simulating it with java based GridSim tool with different application, structures and sizes. The experiments and results show that the scheduling efficiency and execution success ratio of the proposed IGA-RORS outperforms algorithm called GA based Reliability optimized Resource Scheduling algorithm (G-RORS) with and without fault tolerance; it improves the performance of grid system and execute the given application most successfully in the given deadline and budget.

In our future work, we plan to modify our algorithm to enhance its performance on load balancing aspect and to suitable for Cloud computing.

REFERENCES:

[1] I.Foster and C.Kesselman, "The Grid 2: Blueprint for a New Computing Infrastructure", Morgann-Kaufmann, 2003.

[2] Krauter.K, Buyya.R., and Maheswaran.M, "A Taxonomy and Survey of Grid Resource Management Systems for Distributed Computing", *Software- Practice and Experience*, Vol. 32, No. 2, 2002, pp. 135-164.

[3] S.K,Das, D.J.Harvey, and R.Biswas, "Parallel Processing of Adaptive Meshes with Load Balancing", *IEEE Transaction on Parallel and Distributed Systems*, Vol. 12, No. 12, 2001, pp. 1269-1280.

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

[4] Y.S.Dai and G.Levitin, "Reliability and performance of tree-structured grid services", *EEE Transaction Reliability*, Vol. 55, No. 2, 2006, pp. 337-349.

[5] Y.S.Dai, G.Levitin, and K.S.Trivedi, "Performance and Reliability of Tree Structured Grid Services Considering Data Dependence and Failure Correlation", *IEEE Transaction on Computers*, Vol. 56, No. 7, 2007, pp. 925-936.

[6] Y.S.Dai and X.L.Wang, "Optimal Resource Allocation on Grid Systems for Maximizing Service Reliability Using a Genetic Algorithm", *Reliability Engineering and System Safety*, Vol. 91, No. 7, 2005, pp. 1071-1082.

[7] Yuan-Shun Dai, Yi Pam, and Xukai Zou, "A Hierarchical Modeling and Analysis for Grid Service Reliability", *IEEE transaction on Computers*, Vol. 56, No. , 2007, pp.681-691.

[8] F.Berman, R.Wolski, H.Casanova, W.Cirne, H. Dail, M. faerman, S.Figueira, J.Hayes, G. Obertelli, J. Schopf, G. Shao, S. Smallen, N. Spring, A. Su, and D. Zagorodnov, "Adaptive Computing on the Grid Using AppLeS", *IEEE Transaction On Parallel and Distributed Systems*, Vol. 14, No.14, 2003, pp. 369-382.

[9] D.J.Chen, R.S.Chen, and T.H.Huang, "A Heuristic Approach To Generating File Spanning Trees For Reliability Analysis Of Distributed Computing System", *Computers and Mathematics With Application*, Vol. 34, No. 10, 1997, pp. 115-131.

[10] K.S.Trivedi, "Probability and Statistics with Reliability, Queuing and Computer Science Applications", *second ed. John Wiley and Sons*, 2001.

[11] Xiaoyong Tang, Kenli Li, Meikang Qiu, and Edwin H.-M. Sha, "A Hierarchical Reliability-Driven Scheduling Algorithm In Grid Systems", *Journal of Parallel Distributed Computing*, Vol. 72, No. 4, 2012, pp. 525–535.

[12] D.Goldberg, Genetic algorithms in Search, Optimization and Machine Learning. Readig, MA: Addison Wesley, 1989.

[13] Yuan-Shun Dai and Xiao-Long Wang, "Optimal Resource Allocation On Grid Systems For Maximizing Service Reliability Using A Genetic Algorithm", *Reliability Engineering and System Safety*, Vol. 91, No. 9, 2006, pp. 1071–1082.

[14] Way Kuo and Rui Wan, "Recent Advances In Optimal Reliability Allocation", *IEEE Transaction On Systems, Man, and Cybernetics- Part A: Systems and humans*, Vol. 37, No. 2, 2007, pp. 143- 156.

[15] Xiao Qin and Hong Jiang, "A Dynamic and Reliability Driven Scheduling Algorithm for Parallel Real-Time Jobs Executing on Heterogeneous Clusters", *Journal of Parallel Distributed Computing*, Vol.65, No.8, 2005, pp. 885–900.

[16] H.El-Rewini and T.Lewis, "Scheduling Parallel Program Tasks onto Arbitrary Target Machines", *Journal of Parallel Distributed Computing*, Vol. 9, No. 2, 1990, pp. 138-153.

[17] G.Sih and E.Lee, "A Compile-Time Scheduling Heuristic for Interconnection Constrained Heterogeneous Machine Architectures", *IEEE Transaction on Parallel Distributed System*, Vol. 4, No. 2, 1993, pp. 175-187.

[18] H.Topcuoglu, S.Hariri, and M-Y.Wu, "Performance-Effective and Low Complexity Task Scheduling for Heterogeneous Computing", *IEEE Transaction on Parallel Distributed System*, Vol. 13, No. 3, 2002, pp. 260-274.

[19] H.Topcuoglu, S.Hariri, and M.Y.Wu, "Task Scheduling Algorithms for Heterogeneous Machines", *Proceedings of the Heterogeneous Computing Workshop (HCW)*, San Juan, 1999, pp. 3–14.

[20] X. Tang, K. Li, R. Li, and B. Veeravalli, "Reliability-Aware Scheduling Strategy for Heterogeneous Distributed Computing Systems", *Journal of Parallel Distributed Computing*, Vol. 70, No. 9, 2010, pp. 941–952.

[21] Syed Abudhagir.U and Shanmugavel, S.: "Performance Optimized Tree Structured Grid Services Considering Error Rate", *Proceedings of International Conference on Information Management and Engineering (ICIME)*, Kuala Lumbur, April 3-5, 2009, pp.471-474.

[22] Fiaz Gul Khan, Kalim Qureshi, and Babar Nazir, "Performance Fault Tolerance Techniques in Grid Computing Systems", *Computer and Electrical Engineering*, Vol. 36, No. 6, 2010, pp. 1110-1122.

[23] Licheng Jiao and Lei Wang, "A Novel Genetic Algorithm Based on Immunity", *IEEE Transactions on Systems, Man, and Cybernetics -Part A: Systems And Humans*, Vol. 30, No. 5, 2000, pp. 552- 561.

[24] HwaMin Lee, KwangSik Chung, SungHo China, JongHyuk Lee, DaeWon Lee, Seongbin Park, and HeonChang Yua, "A Resource Management and Fault Tolerance Services in Grid Computing", *Journal of Parallel*

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

Distributed Computing, Vol. 65, No. 11, 2005, pp. 1305 – 1317.