A Reliable Schedule with Budget Constraints in Grid Computing

I. Stephie Rachel,
Post-Graduate Student,
Department of Computer Science and Engineering,
Karunya University,
India

Joshua Samuel Raj,
Assistant professor,
Department of Computer Science and Engineering,
Karunya University,
India

V. Vasudevan
Registrar,
Kalasalingam Univeristy,
Srivilliputhur,
India.

ABSTRACT

The application system while executing in a Grid environment may encounter a failure. This phenomenon can be overcome by a reliable scheduler who plays the major role of allocating the applications to the reliable resources based on the reliability requirement of the applications given by the users. The reliability requirement considered in this paper is deadline and budget which is also the quality of service requirement needed for the applications. In this paper, based on deadline and budget as a main factor the tasks are scheduled to the reliable processors.

Keywords: Grid Scheduling, Deadline, Budget, Reliability

1. INTRODUCTION

Grid Computing is a type of distributed computing to solve complex, huge computations where the resources in the remote places are accessed by connecting all the resources together in a network. The resources of computers owned by individuals or by organizations from several countries are connected to form a single, vast super computer. A Grid gathers together resources and makes them accessible in a secure manner to users and applications [1]. Grid computing is needed when there is a necessity for huge computing of data and the data is stored in different institutions. Based on the needs, grid is classified into Private vs. Public, Regional vs. Global, All-purpose vs. Particular scientific problem. The large applications from the users are divided into a set of subtasks and sent to the several resources connected to the main server which is freely available. There are four different classes of Grid users such as end-users of applications, application developers, system administrators, and managers of organizations. After the computations are over at the particular resources, the results are sent back to the main global server. As soon as all the other computations are received by the global server, the result is then provided to the user.

Grid scheduling plays the major role to schedule the tasks onto the processor efficiently. There are three phases in grid scheduling such as Resource discovery, System selection and job execution. When executing the job, the resources should be reliable so that the job can execute successfully. The reliability of the grid system is affected by several factors such as hardware failure, software failure. Reliability of computational hardware, software and data resources that comprise the grid and provide the means to execute user applications and reliability of grid networks for messaging and data transport are important and should be met [2].Ignoring Grid reliability characteristics can lead to reduced application performance, such as schedule length and speedup, due to wasted operations [3]. Reliability of the grid system depends on the failure rate of the processors and the links between them. These failure rates can be derived from Grid resource's profiling, system log, and statistical prediction techniques [4]. Reliability of a system with expect to a task set as the probability that the system can run the task set without any failure [5].

2. RELATED WORKS

The objective of the paper described in [6] to design reliability-driven scheduling architecture to measure system reliability, based on an optimal reliability communication path search algorithm to find the shortest path, and then they introduce reliability priority rank (RRank) to estimate the task's priority by considering reliability overheads. RASD algorithm consists of two mechanism such as a listing mechanism-heuristic for HDC systems and Processor assignment mechanism based on task-duplications. Tasks are assigned to the processors and duplications are employed to reduce their costs which result in reducing the schedule length and improving the system reliability. Two heuristic algorithms such as Minimum cost Match scheduling and Progressive Reliability Maximization Schedule are used in Reliability Driven Task Scheduling Algorithm to improve the system reliability and schedule length. MCMS constructs a bipartite matching graph and schedules tasks to processors according to the min-cost maximum bipartite matching. PRMS uses a progressive relaxation strategy based on a schedule obtained from the ALAP scheduling. After this task is scheduled, mark the node, update the dependence constraint information, system reliability cost and the schedule length, then continue this process until all nodes are marked [7].

Repars and Refine is proposed for Periodic tasks in a heterogeneous system to improve reliability while meeting the time constraints and to enhance the system reliability while being able to tolerate the failures by introducing primary backup scheme respectively. Repars algorithm selects a processor providing the task with the smallest reliability cost which in turn results in the highest reliability for the task. Refine algorithm considers fault-tolerance in addition with reliability of the system. It calculates based on first-fit strategy, the task which reliability cost is small enough to fit into processor [8]. The objective of algorithm RCD is minimizes the schedule length and maximizes the reliability.

RCD is designed so that it no need of extra hardware cost. Advantage of RCD over the non-RCD algorithms in schedulability mainly comes from the variance in tasks' reliability costs among different processors [9].

The objectives of this paper are minimizing the makespan and maximizing the reliability. The two Multi-Objective Evolutionary Algorithms such as Multiobjective Genetic Algorithm (MOGA) and Multi objective Evolutionary Programming (MOEP) with non-dominated sorting are developed which account both for schedule length and the failure probability based on non-dominated ranking for selection. MOEA combines Evolutionary computation and theoretical frameworks of multi-criteria decision making. MOGA is a probabilistic and uses global search technique [10]. In Cost and dependence matrix for grid algorithm (GCDM), the cost of data transfer between the costs. Two matrix such as CT * T matrix and different tasks and dependencies between tasks is modeled as an acyclic directed graph (DAG) to minimize DT * T matrix for data transmission cost matrix between the depending nodes and dependency graph nodes matrix respectively. The purpose of this algorithm is to assign the task to the best resources [11].

The Dynamic and Reliability-Driven Scheduling Algorithm is employed to increase the reliability by minimizing the system reliability cost. First DASAP (schedule As Soon As Possible) and DALAP (schedule As Late As Possible) are used to assign the task to the processors, in which system reliability is not considered then Dynamic Reliability-Cost-Driven (DRCD) Scheduling Algorithm is designed to assign the task to the processors which take reliability cost into account [12]. The objective of RDGS algorithm is to maximize the total number of tasks completing execution based on Communication to Computing Ratio (CCR) which decides the appropriate grid site for scheduling tasks [13].

The selection of heuristic depends on the structure of the heterogeneity among tasks and machines, the optimization requirements, and the arrival rate of the tasks [14]. HRDS is a hierarchical reliability-adaptive scheduler, assigns each task to a processor according to its computation time and reliability overhead. Based on the application needs, global server resource management system assigns the weight of the reliability overhead value to the local server.

3. SYSTEM MODEL

User sends their applications to the server. Grid scheduler splits the application into a subset of tasks or jobs, so that tasks execute efficiently on the processors. The subset of tasks reached for scheduling may be dependent on each other's task or may be independent. If the task is dependent on each other, the output of parent task flows to the immediate child task. The dependent tasks are represented by Directed Acyclic Graph (DAG). DAG represented by nodes and edges. Nodes are represented by N1 to N7 in the diagram. Edges are represented by E1 to E10 in the diagram. Each node has its own computation cost, that is the cost required to execute the task on the processor. Each edge has communication cost, that is the cost required to flow the data from one node to another node. The Directed Acyclic Graph figure is shown in fig 1. Directed Acyclic Graph (DAG) $G = (N, E)$, N is the set of n tasks; E is the edge between the tasks to represent the dependences. The weight W (n) is assigned to task n represents the computation cost and the Weight W ($e_{i,j}$) assigned to edge $e_{i,j}$ represents the communication cost. A node without predecessors, pred (n) is called an entry task. A node without successors, succ(n), is called an exit task. The set of Resources are denoted by $\{R_1, R_2 \ldots R_n\}$.

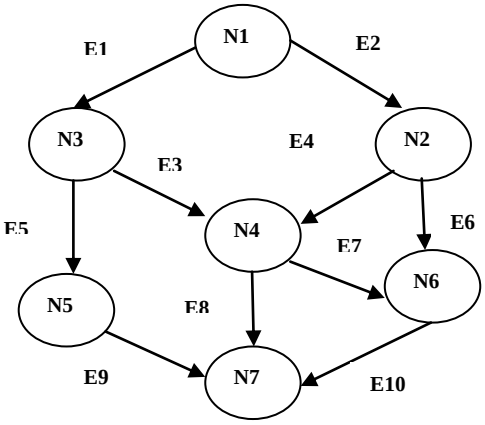

Fig 1: Directed Acyclic Graphs

4. SCHEDULING ALGORITHM

The divided applications are sent as a DAG. The computation and commuications costs of every task is calculated by the Global scheduler. If the user gives a high budget, the tasks are scheduled on to the reliable processors which have a low failure rate. The failure rate is stored in a System log by the Grid Information System(GIS). The Reliabilty probablity of the tasks are calculated to guarentee that each task is successfully executed on the processor and the output succefully reaches its child node. Reliability of the task calculation helps to schedule the tasks on the reliable processor based on the value.

The operational model of the Architecture is as follows

1. Initially user sends the application with the deadline and Budget to the Global server and is divided into subset of dependent tasks by the grid scheduler.
2. GIS in the Global server updates the information about the available resources connected in the network at a fixed interval of time.The information such as Number of resources, cost, capacity of each resource, failure rate of the resources from the history records, communication delay among the resources are saved periodically.
3. Resource Manager in the Scheduler requests the GIS for freely available resources details. Global scheduler maintains the Lookup table.
4. After receiving the information from the GIS, it uses the Lookup table to match the budget of the resources with the value given by the user.
5. Based on the compared value of deadline and Budget the Rank is given to the each task.
6. Based on the Rank value, the algorithm is selected and executed, then the Resource Manager selects the resources and queue the tasks to each resource queue.
7. Dispatcher selects the task from the Resource Queue.

8. Dispatcher submits the tasks onto the respective resources successfully.
9. Finally Scheduler assembles the output of the resources and sends it to the corresponding user.

The Global scheduler selects the algorithm based on criteria given in the table and runs the respective algorithm, so that it can save time and meet the user needs.

Table1 shows the Global scheduler Lookup table. Based on the rank level, it will put the task on the queue and it will add the numberic rank as additional parameter, and execute the tasks on the reliable Resources. The Economic Based Reliable Grid Scheduling Architecture is shown in the fig 2.

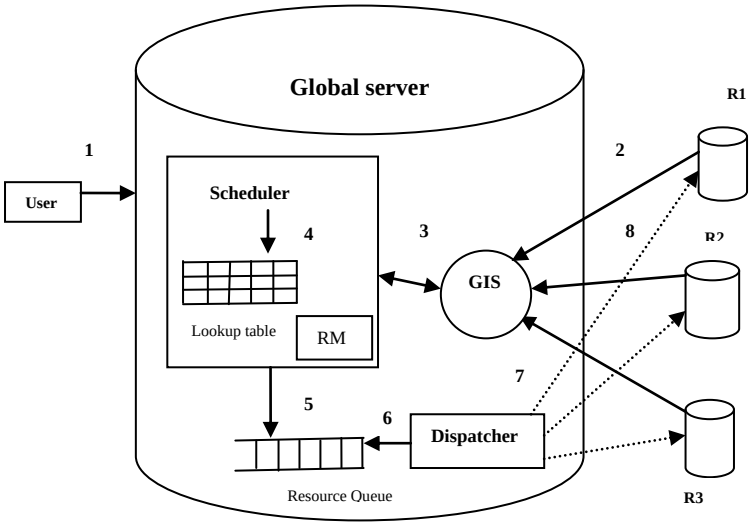

Fig 2: Economic Based Reliable Grid Scheduling Architecture

Table1:Global scheduler Lookup table

Budget	Deadline	Parameter concern	Rank level	Approach
High	Near & hard	Budget-based & deadline based	very high 1<i<2	EEFT & Reliability of the resources
Low	Near & soft	Deadline based	high 2<i<3	EEFT & Reliability of the resourecs
High	Far & hard	Budget based	medium 3<i<4	Reliablity of the resources
Low	Far & soft	____	low	____

4.1 DEFINITION AND ASSUMPTION

The HEFT algorithm selects the task with the highest rank value at each step and assigns the selected task to the processor, which minimizes its earliest finish time by calculating EEFT. Here the Rank is based on the deadline & budget. If the rank level is higher most and Highest then EEFT algorithm is performed first and then followed by calculating reliability of the tasks on the processor. Work done is calculated by Execution time of load of the task on the Resource is

$$Work\ done = \frac{Weight\ of the\ task\ W(t)}{Capacity\ of\ the\ \mathrm{Re}source\ C(R_i)} \quad (1)$$

Earliest Execution Finish Time (EEFT) of the task on the resource is

$$EEFT(n,R) = EEST(n,R) + \frac{Workdone}{Rs} \quad (2)$$

Where EEST is the earliest execution start time of a task on the resources and Rs is the processor clock speed.

The EEFT of the parent node should be greater than than the EEST of the child node,So that the output data can be flown from parent to child node. Assume that resourecs are in the different locations. So there must be the delay in the communications among the procesors. For each task, reliabilty of the task is calculated in percentage by reliabilty of the task execution on the resources successfully without resources failure rate and delay between resources.

$$\mathrm{Re}l(n) = Workdone*(1-fR_i)*(1-delay)*100 \quad (3)$$

Where fR_i denotes failure rate of the i^{th} resource R. Both failure rate of the resources and delay among the virtually connected resources is from system log.

5. PERFORMANCE EVALUTION

Using Budget and Deadline as a Main factor the Graph is drawn shown in fig 3 and fig 4. In Grid simulator Jobs are represented as Gridlets. Each Gridlet consists of job length, the size of input and output files and job owner id,

along with that user have to give deadline(D-factor) and budget(B-factor).

If the deadline is relaxed then the priority of the task is low , perhaps the gridlets placed in the final positions in the resource queue. If the deadline is tight the priority is higher, executed earlier so that the task doesnt miss the deadline.

1. Grid Information System:

for each fixed clock rate
do
if GIS receive signal
Updates *GIS* with resource information
{R_1, R2…R_N} details
Else
Update Delay rate or failure rate fp_i /disconnected
End if
End

2. Grid Scheduler:

for each fixed clock rate
do
Updates *RM* with freely available resource
{Rf_1,Rf_2…Rf_N} details
end

- Divide applications into subset of tasks
  DAG G=<V,E>
- Set the computation costs of tasks,
  communication costs of edges
- Compare the user B-value with the
  available resource B-value
- Sort the deadline
- Rank the tasks and choose the algorithm
- based on the criteria in the *Lookup table*

if Rank Level i < 3 where i=1
Get the sorted Deadline array D[i]
for each available Resources R
do
Calculate workdone using Eq 1
Calculate EEFT(n,R_i) using the Eq 2
end
Select the suitable resources and put the tasks
in corresponding *Resource Queue* {Rq_1,Rq_2…Rq_N}
end if

if Rank Level i < 4 where i=3
for each available Resource R
do
Compute the Rel(n) using Eq3
end
Select the suitable resource and put the tasks in
corresponding *Resource Queue* {Rq_1,Rq_2…Rq_N}
else
Assign the tasks on the any resource available
end if

3. Dispatcher
Dispatch the task from the Resource Queue
{Rq_1,Rq_2…Rq_N} to the respective resources

Algorithm: Economic based Reliable scheduling algorithm

In our experiment the resources are 2000GB, 2500GB, 6000GB, 9000GB and 12000GB. As the capacity of the resources increases, the speed and budget of the processor will be low. If the resource R1 has low capacity it should have high speed, so it is capable to execute faster and finish the execution before the deadline. The Execution start time of the task depends on the execution finish time of the previous task executed on the resources scheduled by the Grid scheduler. In dependency tasks, the execution start time of the child tasks should be after the execution finish time of the parent tasks.

The execution start time of the tasks is assumed based on these criteria. In this graph the Tasks which have near deadline is executed first, but also based on the budget of the application too.

After the scheduler executes the EEFT algorithm, it will execute the Reliability of the task on the resources. If

the user gives high budget value, it is possible to select the R1 based on the execution time, but the scheduler also check the Reliability of the resources and schedule the tasks to the freely available processors. In a scenario the budgets are given to the applications by the users as 0, 900, 1400, 1500 and 2500 in dollars as shown in the fig 4. If the user wants to execute the application freely, the scheduler will execute the applications on the resources which have low budget value. The reliability of the tasks is calculated in percentage.

The Deadline Vs EEFT is illustrated in the fig 3 and The Reliability Vs Budget of the application is illustrated in the fig 4. The EEFT is diagonally increases from left to right in the fig 3.Thus the deadline and reliability based scheduling is done using Economic based Reliable scheduling algorithm.

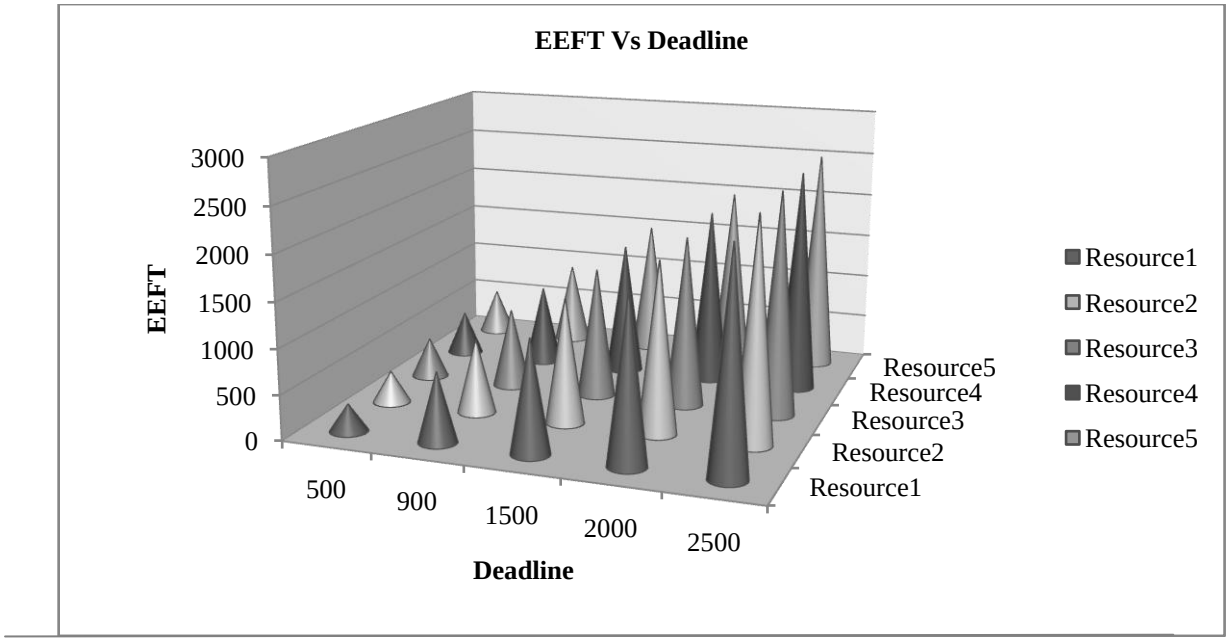

Fig 3: Deadline Vs EEFT

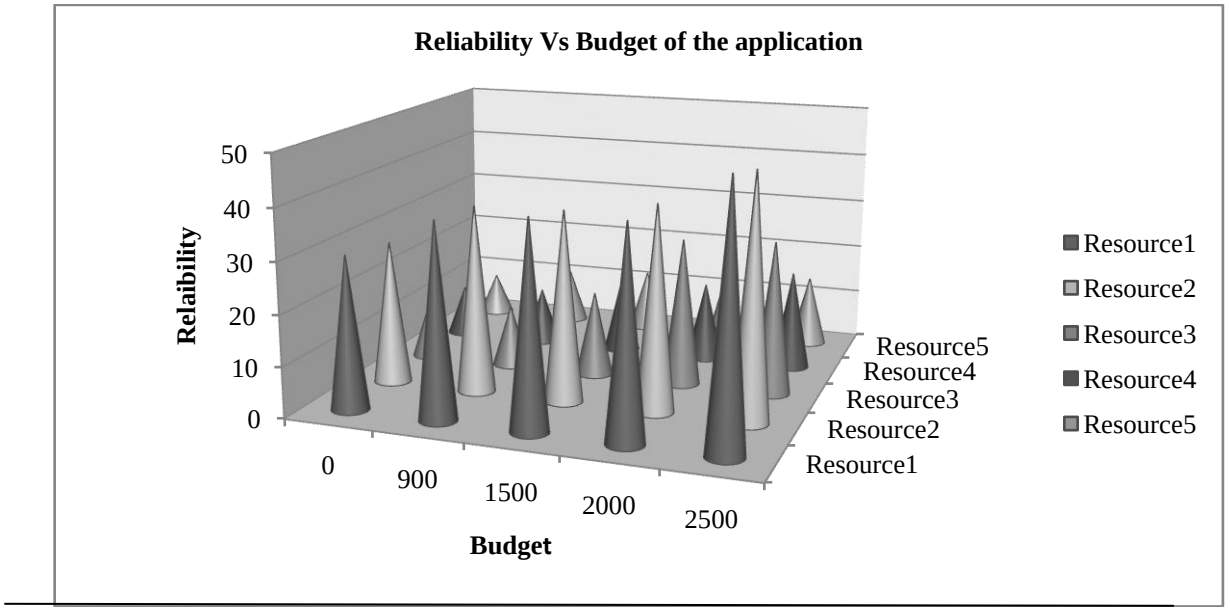

Fig 4: Reliability Vs Budget of the application

6. CONCLUSION

In this paper, both deadline and budget are used as a main factor to schedule the tasks on reliable processors. Based on the user requirement, the algorithm is executed to achieve the reliability and it satisfies the user requirement. This approach saves the running time of the algorithm. In the future, further user requirement is planned to be taken into account for reliability. Thus the deadline and reliability based scheduling is done using Economic based Reliable scheduling algorithm.

7. REFERENCES

[1] Grimshaw, A. What is a Grid? Grid Today, 1(26), 2002.

[2] Christopher Dabrowski, "Reliability in grid computing System" Concurrency & computation practice & experience, 2009.

[3] A. Dogan, F. Ozguner, Biobjective scheduling algorithms for execution time reliability trade-off in heterogeneous computing systems, Comput. J. 48 (3) (2005) 300–314.

[4] Xiaoyong Tang, Kenli Li, Meikang Qiu, Edwin Hsing-Mean Sha: A hierarchical reliability-driven scheduling algorithm in grid systems. J. Parallel Distrib. Comput. 72(4): 525-535 (2012)

[5] S.Srinivasan and N.K Jha, "Safety and Reliability Driven Tasks Allocation in Distributed Systems",IEEE Trans.Parallel and distributed system, Vol.10 No3, Mar.1990.

[6] X. Tang, K. Li, R. Li, B. Veeravalli, "Reliability-aware scheduling strategy for heterogeneous distributed computing systems",J. ParallelDistrib. Comput. 70 (9) (2010) 941–952.

[7] Y. He, Z. Shao, B. Xiao, Q. Zhuge, Edwin Sha, "Reliability driven task scheduling for Heterogeneous systems", in: The International Conference on Parallel and Distributed Computing and Systems, 2003, pp. 465–470.

[8] Wei Luo, Xiao Qin, Kiranmai Bellam, "Reliability-Driven Scheduling of Periodic Tasks in Heterogeneous Real-Time Systems", 21st International Conference on Advanced Information Networking and Applications Workshops, 2007 (1) 778- 783

[9] Xiao qin, hong jiang ,Changsheng xie And Zongfen han , "Reliability-driven scheduling for real-time tasks with precedence constraints in heterogeneous systems", 12 th International Conference Parallel and Distributed Computing and Systems,2000

[10] P.Chitra, P.Venkatesh and R.Rajaram, "Comparison of evolutionary computation algorithms for solving bi-objective task scheduling Problem on heterogeneous distributed computing systems", on Indian Academy of Sciences, 2011 (36) 167-180.

[11] Amir M Bidgoli, Zahra Masoudi Nezad, "A new scheduling algorithm design for grid computing tasks", 5th SASTech 2011, Khavaran Higher- education Institute, Mashhad, Iran. May 12-14.

[12] Xiao Qin and Hong Jiang. A Dynamic and Reliability-driven Scheduling Algorithm for Parallel Real-time Jobs on Heterogeneous Clusters. Journal of Parallel and Distributed Computing (JPDC), Vol. 65, No. 8, pp 885-900, August 2005.

[13] Kovvur Ram Mohan Rao, Ramachandram , Vijaya Kumar Kadappa and Govardhan , "A Reliable Distributed Grid Scheduler for Independent Tasks", IJCSI International Journal of Computer Science Issues, Vol. 8,Issue 2, March 2011.

[14] Muthucumaru Maheswaran, Shoukat Ali and Howard Jay Siegel, Debra Hensgen, Richard F. Freund, "Dynamic Mapping of a Class of Independent Tasks onto Heterogeneous Computing Systems" Journal of Parallel and Distributed Computing 59, 107_131 (1999)

AUTHOR'S PROFILE

Stephie Rachel I pursuing her M.Tech in Computer Science and Engineering from the department of computer science in Karunya University received her B.Tech. in 2011 from P.S.R Engineering College under Anna University of Technology Tirunelveli in Information Technology.

Joshua Samuel Raj, working towards the Ph.D degree in grid scheduling received his B.E degree from PETEC under Anna University and M.E degree from Jeya College of Engineering under Anna University. His areas of interest include Grid computing, Mobile Adhoc Networking, Multicasting and so forth

Dr. V. Vasudevan, Registrar of Kalasalingam univeristy received his Ph.D degree from Madurai Kamaraj University in the year 1992. He is the Project Director for the Software Technologies Group of TIFAC Core in Network Engineering. His areas of interest include Grid computing, Mobile Adhoc Networking, Multicasting and so forth