DYNAMIC SCHEDULING OF DATA USING GENETIC ALGORITHM IN CLOUD COMPUTING

A.Kaleeswaran[1], V.Ramasamy[2], P.Vivekanandan[3]
[1, 2 & 3] Department of Computer Science and Engineering, Park College of Engineering and Technology, Coimbatore, India

ABSTRACT

Cloud Computing is the utilization of pool of resources for remote users through internet that can be easily accessible, scalable and utilization of resources. To attain maximum utilization of resources the tasks need to be scheduled. The problem in scheduling is allocating the correct resources to the arrived tasks. Dynamic scheduling is that the task arrival is uncertain at run time and allocating resources are tedious as several tasks arrive at the same time. To avoid this scheduling problem, Genetic Algorithm is used. Genetic algorithm is a heuristic method that deals with the natural selection of solution from all possible solutions. Using genetic algorithm the tasks are scheduled according to the computation and memory usage. The tasks are scheduled dynamically. The execution time is reduced by parallel processing. The scheduled data is stored in cloud. By using GA we obtain global optimization.

KEYWORDS **-** *Cloud computing, Resource utilization, Dynamic scheduling, Genetic Algorithm and Optimization.*

I. INTRODUCTION

Cloud computing provides on-demand network access to a shared pool of resources. Cloud computing ensures access to virtualized IT resources that are present at the data center, and are shared by others. The data stored in Cloud are simple to use, and are paid for the usage and can be accessed over the internet. These services are provided as a service over a network, and are accessible across computing technologies, operations and business models. Cloud enables the consumers of the technology to think of computing as effectively limitless, of minimal cost, and reliable, we need not know about how it is constructed, its working, its operation, or where it is located.

Cloud Computing reduce or avoid capital expenditure by renting third party services. Customers use resources as services and only pay for what they use. Like electricity and water services the customers pay only for their consumption. Cloud provides to its users, massive storage and maintenance of large volumes of data in reality. Workload is balanced efficiently, since the cloud can scale dynamically.

Scheduling is used for distributing resources among parties which simultaneously and asynchronously request them. For the purpose of scheduling, scheduling algorithms are used. Scheduling algorithms are used mainly to minimize resource starvation and to ensure fairness amongst the parties utilizing the resources. Scheduling deals with the problem of which resources needed to be allocated for the received task. Many scheduling algorithms are currently available. Scheduling is the process of converting an outline plan for a project into a time-based graphic presentation using time constraints. Scheduling through the Task Scheduler enables us to automatically perform routine tasks on a chosen schedule. The Task Scheduler does this by monitoring whatever criteria we choose to initiate the tasks and then execute the task when the criteria are met. With Scheduled Tasks, we can schedule any script, program, or document to run at a time that we specify when creating the task. The task scheduling problem optimizes the overall performance of the application and provides assurance for the correctness of the result

This paper is organized as follows: Section II presents the Literature Review, Section III explains the Proposed System, Section IV presents Methodologies, and Section V gives the Implementation, Section VI describes the Analysis of the model and Section VII future work and Section VIII gives the conclusion of the paper.

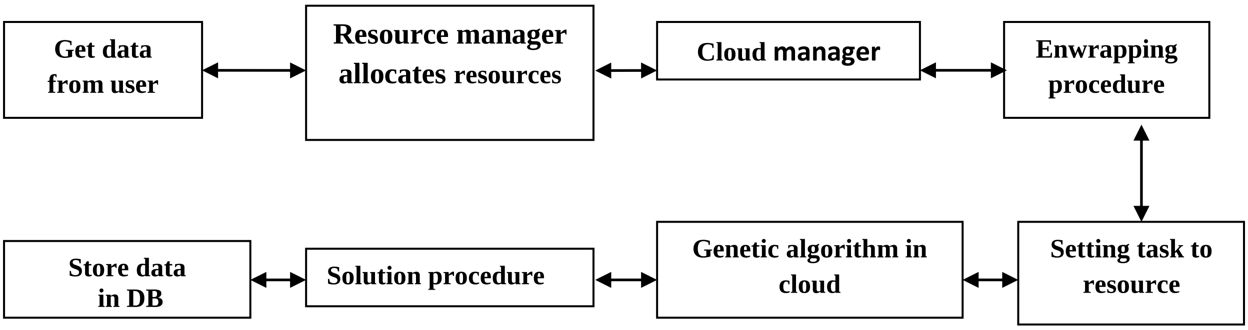


Figure1. Architecture diagram of scheduling using GA

II. RELATED WORK

Scheduling performed using several algorithms are studied in the literature recently. In [2], the author proposes an image caching mechanism to reduce the overhead of loading disk image in virtual machines. The author in [3] presents a dynamic approach to create virtual clusters to deal with the conflict between parallel and serial jobs. In this approach, the job load is adjusted automatically without running time prediction. In [4], the author describes a suspend/resume mechanism is used to improve utilization of physical resource. The overhead of suspending/resume is modeled and scheduled explicitly. In [5] present a planner-guided strategy for multiple workflows. It ranks all ready tasks and decides which task should be scheduled first. However if there are new lower rank workflows coming continuously, the higher rank task will not be scheduled to execute. In a massive scalable cloud, this situation will become true. This algorithm only considers the execution time.

The author in [6] describes the Minimum Execution Time (Met) assigns each job in arbitrary order to the nodes on which it is expected to be executed fastest, regardless of the current load on that node. Met tries to find good job-node pairing, but because it does not consider the current load on a node it will often cause load imbalance between the nodes and not adapt application in the heterogeneity computer systems. On scheduling in grid computing environment, genetic algorithm could be adopted to solve the NP-hard problem. The author in [7] represents an extensive study on the usefulness of GAs for designing efficient Grid schedulers when make span and flow time are minimized. A first-come-first-served and a Genetic algorithm are used for the load balancing strategy [8]. It gives GA based task scheduling for optimum decision.

The author in [9] introduces the Task and Virtual Machine (TVM) scheduler, which schedules VMs in cloud and task on VMs. By scheduling both, task and VMs, it is possible to define virtual computing systems that go beyond the limitation imposed by the availability of resources. The Virtual machines are instantiated in every host in the network prior to the execution of the application. Tasks composing application have requirements that can limit the choice of resources for execution.

III. PROPOSED SOLUTION

A. Scheduling Model

The objective of the system is to have maximum utilization of resources and to reduce the execution time. The scheduler instantiates one task per virtual machine, assuring that at any given time, every host processing will execute one virtual machine instance. The nodes are given by a cluster. Each cluster has an independent master task scheduling node and the slave task assigned node which derives from a cluster under the control of master node. Even though the nodes are from the same cloud they might have different hardware, scheduling is done without flaws. The master node is responsible for scheduling all the tasks, monitoring their implementation, re-run the task that has failed, or disposing of errors. The master node collects information about the nodes that are participating, their memory and processing capacity and transmission rate. The slave node is responsible for implementing the task assigned by the master node. Upon receipt from the assignment of the master node, the slave node starts to find a suitable computing node. Data is distributed through these nodes. It considers the work load on each node. Some nodes may always get busy and some nodes may be idle. The proposed system balances the computation of nodes equally.

The system has enwrapping procedure and solution procedure. The enwrapping procedure phase is to enwrap computing resources into cloud computing system in which a computing resource is semantically parsed and enwrapped as a node service. The Solution Phase is to discretize the input user's computational problems to be series of services through a semantic translator. This series is sent to a solver to be solved and the result is sent back to semantic translator.

When some services are present the tasks are waiting in the queue. The scheduling will recomputed the tasks that are currently present and sort all the tasks. The scheduling is done by taking the first task from the queue and allocating that task to the resource that will best fit using the Genetic Algorithm and begins its processing. When the scheduling is done finally the data is stored in cloud.

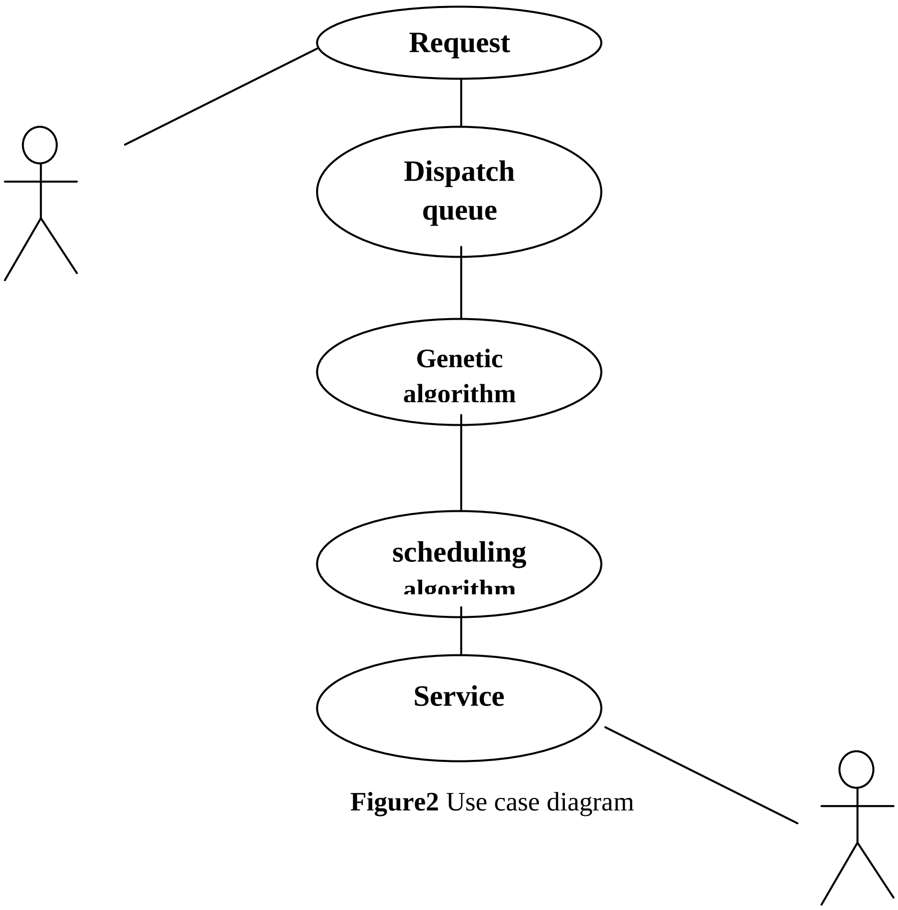


Figure2 Use case diagram

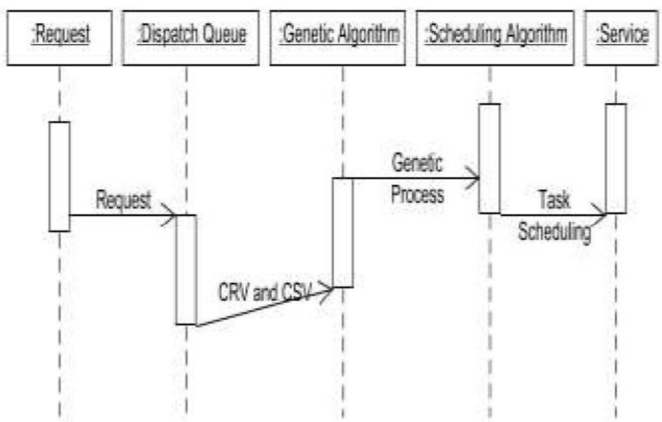


Figure3. Sequence Diagram

B. Algorithm Process

1) Initial Population

When Genetic Algorithm is used to solve problems, initial population provides solutions to the problem. The set of solutions that are possible is taken as the initial population. These solutions are considered as chromosome. The chromosomes in the initial population are generated randomly using the symbol and these terminals are chosen to solve the particular problem. Initial population performs transformation of solved problem to 2Dimensional solution space. Here every chromosome is considered to be string of bits.

Step 1: Initialization Set POP = popsize

Step 2: Execution Set ChromosomeNumber=0

Step 3: If Chromosome Number=popsize
Stop the process
Display the initial population Else
ChromosomeNumber=ChromosomeNumber+1 and set POP[ChromosomeNumber]

Step 4: Set geneNumber=0

Step 5: If geneNumber=n goto Step3 Step 6: setgeneNumber=geneNumber+1

Step 7: Execute the process and select the values such that n>geneNumber

2) Fitness function

Selecting a suitable fitness function is used to design successful Genetic Algorithm. It evaluates how the selected function meets the objective of the problem. GA evaluates each chromosome by fitness function. Fitness function is used for the measurement of effectiveness of the solution according to the given objective. They help to know which chromosomes retain in the population. Incorrect fitness function may lead to delay in process. When the fitness function is larger it meets the QOS requirements of that task.

$$F=\min \{\max \{ck\} + \sum f(di)\}$$

3) Selection

Selection operator is used to select among the given chromosomes of current population for inclusion to next population. It is mainly used to find the best fit individual. Each chromosome has equal probability to its score by the sum of all other chromosome probability. These individuals create next generation. Here range selection is used. Every individual is given a rank based on the fitness function. The individual that is most fit gains more preference when using this method.

Step1: Randomly choose certain chromosomes

Step2: Compare the values with other chromosomes and select the chromosomes for next generation

The value obtained in the fitness function is not directly proportional. It avoids premature stagnation.

4) Crossover

Crossover operator pairs all the chromosomes. It is used to combine two chromosomes to produce next generation chromosomes. It is used for bringing new chromosomes by the mixture of parent chromosomes. Single point crossover is used as only one crossover point is present. In this single crossover point, at the locus, swapping the remaining alleles from parents to others takes place. The operation is performed to select the chromosome. If crossover operation is not performed then the new generation resembles its parents. Bad chromosome is C1 and good chromosome is C2.

For values ranging from $f<=f^{-1}$
C=fC1
For values ranging from $f>=f^{-1}$
$C=c1-\{(c1-c2)/f_{max}-f\}$

5) Mutation

Mutation performs the permutation of existing chromosomes. It is used to maintain the genetic diversity from one generation to next generation. It provides new gene values added to the gene pool. Mutation provides small alterations at each individual. It is used for finding new points in search space hence the population variation is maintained. They provide functions on the chromosomes produced by crossover. They help to overcome trapping at local maxima. Bad chromosome is M1 and good chromosome is M2.

For values ranging from $f<=f^{-1}$
$M=fM1$
For values ranging from $f>=f^{-1}$
$M=M1-\{(M1-M2)/f_{max}-f\}$

The procedure is:

Step 1: Calculate the mutation for every chromosome
Step2: Chromosome is chosen randomly using selection process, set it as POP[i]
Step 3: Generate a new value and store it as POP [new]
Step 4: Calculate the value of CFitnessfunc[new]
Step 5: IF CFitnessfunc[new]>=CFitnessfunc[i] POP [new] is added to the population

6) Termination Algorithm

The selection process is able to copy the chromosome that is having the maximum fit in the given population. This process is repeated for 20 generations. This helps to obtain the highest fit value. The termination depends on the size of the chromosomes that has been sorted.

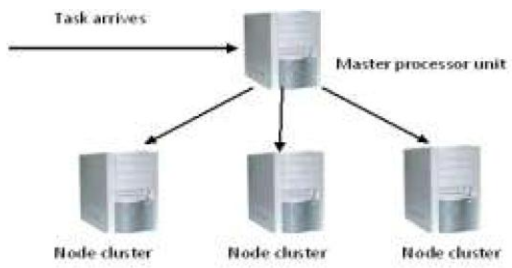


Figure 4: Representation diagram when using Ubuntu

IV. METHODOLOGIES

1) Creation of Private Cloud

Cloud creation is done using Ubuntu. Through Eucalyptus the private cloud creation is done. One system acts as a front end that contains cloud controller, cluster cloud, walrus controller and storage controller. The system that comes under front end are called node and they contain node cluster. They are used to run instances. The cloud controller provides interface to interact with the user cloud. Walrus storage is used for storing and accessing the stored data. Cluster controller is used to manage the virtual network. Cloud controller instructs the node controller and its work is to interact with the operating system

2) Working with Service Node

The basic element is the service node. It contains the enwrapping procedure and the solution procedure. In the enwrapping procedure the resources are enwrapped to the cloud computing environment. In the solution procedure, with the help of semantic translator various problems are separated as services. This is solved by solver.

3) Encoding

Encoding is used to perform the selection of the solution for a problem such that it corresponds the genetic algorithm solution space. It is used for the up gradation of one form to another. Binary encoding is used in this procedure. It provides many chromosomes that are possible from the given alleles.

4) Assigning tasks to resources

The tasks are classified based on heterogeneous system. The tasks are aperiodic. Their arrival times are not known. The tasks are nonpremptive. Tasks are independent to each other. It provides shared access where it allows other tasks to share the resources.

5) Using Genetic Algorithm for allocation

The tasks that are arrived are encoded using binary encoding, and the obtained value is compared with the fitness function. If it does not match then the selection operation is performed. The value obtained from selection is under single point crossover where new generation occurs. Finally mutation occurs where permutation of values occur and is compared with the fitness function.

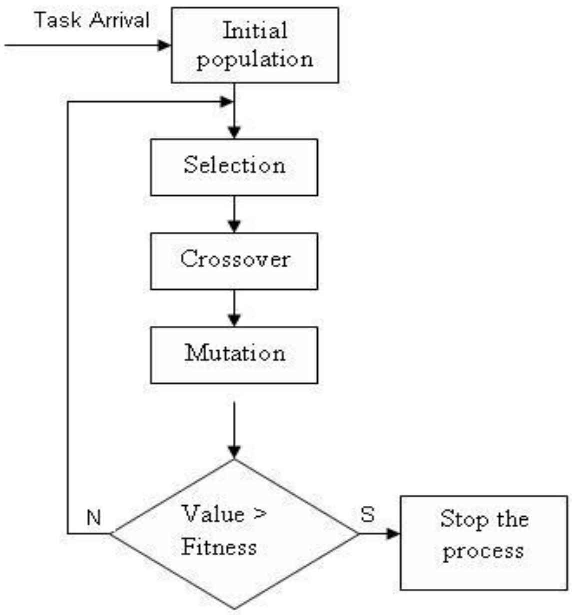


Figure 3: Workflow diagram of GA

6) Storing data in Cloud

Once the tasks are scheduled and resources are allocated the data is stored on Cloud. The stored data gets replicated as it's free from loss of data. This data could be accessed anywhere and anytime. The data is stored persistently. It is independent of virtual machine location and state.

V. IMPLEMENTATION

In the implementation of private cloud the Ubuntu Enterprise Cloud is installed in machined that are used for cloud operation. The other ways of implementation is through Eyeos or by using Open Nebula. The installation of Ubuntu helps us to connect to other systems where their IP address is displayed. We just include them by selecting their IP address. Once the user sends the data through cloud, a centralized scheduling scheme is used where the master processor unit is used for the collection of tasks, which sends these tasks to other processing units. Dispatch Queue is maintained for each processor.

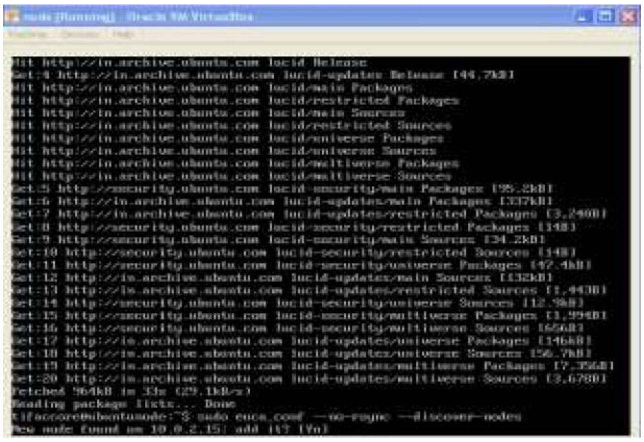
The slaves receive the command from the master processor and checks for the resources that are currently available and that which has the desired memory requirements. The various process units communicate through these dispatch queues. Thus tasks are executed. When the task is being executed the newly arrived task is put in the dispatch queue. Master processor unit works parallel, executing the task and scheduling the newly arrived task. The encoding and the decoding functions are performed to have security over stored data. The scheduled data is finally stored in cloud. User access the cloud to retrieve the scheduled data.

VI. RESULT

Setting up a private cloud is the work done. With the help of Eucalyptus software node discovery is done and their respective IP addresses are displayed. Mulitenancy is supported when using Ubuntu.

This shows node discovery

.



VII. CONCLUSION & FUTURE WORK

Cloud computing system is used to scale the application by maximizing the concurrency and using the resources efficiently. The algorithm takes time utilization and resource utilization into consideration and hence results in high signification. The experiments need to be done for processors at various platforms and that reduces the execution time. In future reducing the solution space must be reduced. Optimization should be done on shared resources, statelessness, partitioning the database and resource utilization. This helps to combine various resources and tasks

REFERENCE

[1]. Chenhong Zhao, Shanshan Zhang, Qingfeng Liu, Jian Xie, Jicheng Hu, "Independent task scheduling based on Genetic Algorithm in Cloud Computing" , 978-1-4244-3693-4/09/$25.00 ©2009.

[2]. W. Emeneker and D. Stanzione, "Efficient Virtual Machine Caching in Dynamic Virtual Clusters." In SRMPDS Workshop, ICAPDS, 2007.

[3]. N. Fallenbeck, H. Picht, M. Smith, and B. Freisleben, "Xen and the art of cluster scheduling," in Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing, 2006.

[4]. B. Sotomayor, R. Llorente, and I. Foster, "Resource Leasing and the Art of Suspending Virtual Machines," in 11th IEEE International Conference on High Performance Computing and Communications, pp. 59–68.

[5]. Zhifeng Yu and Weisong Shi, "A Planner-Guided Scheduling Strategy for Multiple Workflow Applications," icppw, pp.1 -8, International Conference on Parallel Processing - Workshops, 2008.

[6]. LeeCY ,Piramuthu S.Tsai YK, "Job shop Scheduling with a genetic algorithm and machine learning" Inr J.Pred Res.1997 35(4):1171-1191.

[7]. J. Carretero, F. Xhafa, "A genetic algorithm based schedulers for grid computing systems," International Journal of Innovative Computing, Information and Control, vol.3, no. 6, Dec. 2007.

[8]. H.J. Braun, T. D.and Siegel, N. Beck, L.L. Blni, M. Maheswaran, A.I. Reuther, J.P. Robertson, M.D. Theys, and B. Yao, "A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems," Journal of Parallel and Distributed Computing, vol. 61, no. 6, pp.810–837, 2001.

[9]. Saswati Sarkar "Optimum Scheduling and Memory Management in Input Queued Switches with Finite Buffer Space", IEEE 1373, 2003.

[10]. Xu Wang; Beizhan Wang; Jing Huang, "Cloud computing and its key techniques", IEEE International conference on Computer Science and Automation Engineering (CSAE), 2011, Shanghai, pp. 404 410, 2011.

[11]. Hao Yin , Huilin Wu , Jiliu Zhou, An Improved Genetic Algorithm with Limited Iteration for Grid Scheduling, Proceedings of the Sixth International Conference on Grid and Cooperative Computing, p.221-227, August 16-18, 2007.

[12]. Guo G., Ting-lei H., Shuai G., "Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment", IEEE International Conference on Intelligent Computing and Integrated Systems (ICISS), 2010, Guilin, pp. 60--63, 2010 .

[13]. Chenhong Zhao , Shanshan Zhang , Qingfeng Liu , Jian Xie , Jicheng Hu, Independent tasks scheduling based on genetic algorithm in cloud computing, Proceedings of the 5th International Conference on Wireless communications, networking and mobile computing, p.5548-5551, September 24-26, 2009, Beijing, China .
[14]. Abdulal W., Jadaan O. A., Jabas A., Ramachandram S., "Genetic Algorithm for Grid Scheduling using Best Rank Power", IEEE World Congress on Nature & Biologically Inspired Computing, 2009. NaBIC 2009, Coimbatore, pp. 181--186, 2009 .
[15]. Wang J., Duan Q., Jiang Y., Zhu X., "A New Algorithm for Grid Independent Task Schedule: Genetic Simulated Annealing", IEEE World Automation Congress (WAC), 2010, Kobe, pp. 165--171, 2010 .
[16]. Verma R., Dhingra S., "Genetic Algorithm for Multiprocessor Task Scheduling", IJCSMS International Journal of Computer Science and Management Studies, Vol.1, Issue 02, pp. 181--185, 2011.
[17]. Jia Yu and Buyya R., "A Budget Constrained Scheduling of Workflow Applications on Utility Grids using Genetic Algorithms", IEEE Workshop on Workflows in Support of Large-Scale Science, 2006. WORKS '06, Paris, pp. 1--10, 2006.
[18]. Mocanu, Maria E., Mihai F., Ionut A. M. and Nicolae T., "Cloud Computing - Task Scheduling based on GeneticAlgorithms", 2012 IEEE International Systems Conference (SysCon), Vancouver, BC, Canada, pp. 1--6, 2012 .
[19]. Xue Zhang; Wenhua Zeng, "Grid Workflow Scheduling Based on Improved Genetic Algorithm", IEEE International Conference on Computer Design and Applications (ICCDA), 2010, Qinhuangdao, pp. V5-270--V5 273, 2010.

A.Kaleeswaran[1] received his B.Sc (Physics) and MCA from Bharathiar University, Coimbatore and M.E (SE) in Anna University Coimbatore. He has four years of teaching experience. He is now working as a Assistant Professor in Department of Computer Science and Engineering in Park College of Engineering and Technology, Coimbatore. He has many publications to his credit in various international conferences and journals. He has attended a number of conferences, seminars and workshops His research interest is mainly focused on Data mining and Image Processing.

V.Ramasamy[2] received his B.E from Anna University, Chennai and M.E in Anna University Tiruchirappalli. He has four years of teaching experience and one year of industry experience. He is now working as a Assistant Professor in Department of Computer Science and Engineering in Park College of Engineering and Technology, Coimbatore. He has guided many undergraduate and post graduate students. He has attended a number of conferences, seminars and workshops His research interest is mainly focused on Data mining and Network.

P. Vivekanandan[3] is currently working as a Professor, Department of Computer Science and Engineering, Park College of Engineering and Technology, Coimbatore, Tamilnadu, India. He has more than twelve years of teaching experience. He obtained his B.E (Computer Science and Engineering) from Bharathiar University, Coimbatore, India and his MTech (Distributed Computing Systems) from Pondicherry University, Pondicherry, India.and his Ph.D from Anna University Chennai. At present he is also a research scholar of Anna University, India. His research interests include Knowledge Discovery and Data Mining, Soft Computing and Distributed Computing. He has published many research papers in National/International Conferences and Journals. He has attended several seminars and workshops in the past ten years. He has also organized several symposiums and workshops. He has guided more than 20 UG projects. He is a life member of ISTE and also a member of Computer Society of India.