

Dynamic Scheduling of Data Using Genetic Algorithm in Cloud Computing

S.Ravichandran¹, E.R.Naganathan²

¹Research Scholar, Department of Computer Science, Bharathiyar University, Coimbatore

²Prof. & HOD, Department of Computer Science & Engineering, Hindustan University

Email: ravi17raja@gmail.com & karpagaravi15@gmail.com, ernindia@gmail.com

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 to obtain global optimization.

Keywords - Cloud computing, resource utilization, dynamic scheduling, Genetic Algorithm, 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, to 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 to choose to initiate the tasks and then execute the task when the criteria are met. With Scheduled Tasks, can schedule any script, program, or document to run at a time that to 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.

Genetic algorithms are a part of evolutionary computing, which is a rapidly growing area of artificial intelligence. Algorithm is started with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions which are selected to form new solutions (offspring) are selected according to their fitness - the more suitable they are the more chances they have to reproduce. To use a genetic algorithm, you must represent a solution to your problem as a *genome* (or *chromosome*). The genetic algorithm then creates a population of solutions and applies genetic operators such as mutation and crossover to evolve the solutions in order to find the best one(s). The algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm randomly selects individuals from the current population and uses them as parents to produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution.

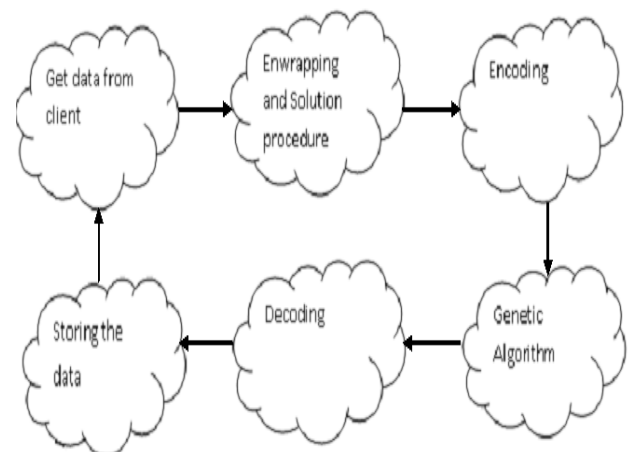


Figure1. Architecture diagram of scheduling using GA

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.

II. RELATED WORK

Scheduling performed using several algorithms are studied in the literature recently. In, the author proposes an image caching mechanism to reduce the overhead of loading disk image in virtual machines. The author in 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, the author describes a suspend/resume mechanism is used to improve utilization of physical resource. The overhead of suspending/resume is modelled and scheduled explicitly. In 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 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 represents an extensive study on the usefulness of GAs for designing efficient Grid schedulers when makespan and flow time are minimized. A first-come-first-served and a Genetic algorithm are used for the load balancing strategy. It gives GA based task scheduling for optimum decision. The author in 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 recompute 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.

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 ChromosomeNumber=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 $n > \text{geneNumber}$

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.

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

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.

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

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

The procedure is:

- Step 1: Calculate the mutation for every chromosome
- Step 2: 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

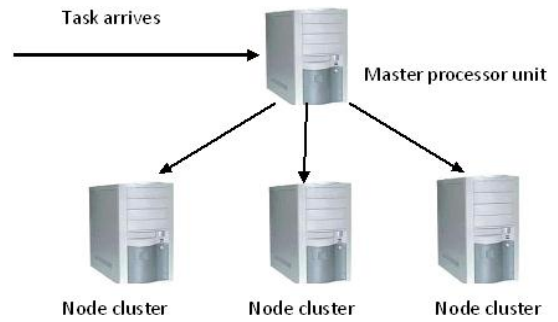


Figure 2: Representation diagram when using Ubuntu

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.

IV. METHODOLOGIES

A. 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.

B. 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.

C. Encoding

Encoding is used to perform the selection of the solution for a problem such that it corresponds to 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.

D. 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 nonpreemptive. Tasks are independent to each other. It provides shared access where it allows other tasks to share the resources.

E. 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. Project planning is a notoriously difficult task. Typically, a project planner has to cope with all the complexity of combinatorial optimisation problems, along with restricted resources. The main algorithm used is a variant of a standard genetic algorithm with tournament selection and one-point crossover. A standard genetic algorithm works by generating new "chromosomes" (solutions) by mutating or combining existing chromosomes, adding these offspring to the original population, and then sorting the population according to a fitness function and removing the worst chromosomes to keep the population size constant.

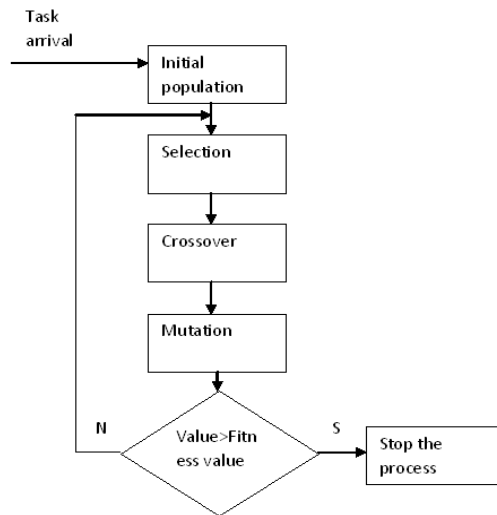


Figure 3: Workflow diagram of GA

F. 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. The 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

Comparing to other algorithms for scheduling, Genetic Algorithm is used to obtain optimum solution and the average execution time is reduced. The installation and updating of changes in software is unnecessary when using cloud. Various changes to the application stored in cloud are done automatically. Multitenancy allows various operating systems to run in single machine. The efficiency of the solution can be varied by altering the fitness function. As this system allows shared allocation of resources the execution is done in parallel and the processing time is reduced.

VII. CONCLUSION

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. By using Genetic Algorithm for resource allocation optimum solution is obtained. 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

REFERENCES

- [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. Emenecker 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. Blini, 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] Hsiao-Lan Fang. *Genetic Algorithms in Timetabling and scheduling*. (Ph.D. thesis) Department of Artificial Intelligence, University of Edinburgh, 1994

- [11] Darrell Whitley. *GENITOR: a different genetic algorithm*. In Proceedings of the Rocky Mountain conference on Artificial Intelligence. Denver, Colorado.
- [12] M Gröbner, P Wilke. *Optimizing Employee Schedules by a Hybrid Genetic Algorithm*. E.J.W Boers et al.(Eds.) EvoWorkshop 2001, LNCS 2037, pp. 463-472. Springer-Verlag, Berlin, Heidelberg

Author Profile:

S. Ravichandran– S. Ravichandran is an Associate Professor in Department of Information Technology at Sri Krishna Engineering College. He received the Master of Computer Applications in Bharathidhasan University at 1996, he received the Master of Philosophy in Computer Science in Madurai Kamaraj University at 2007, he received the Master of Engineering in Computer Science and Engineering from Anna University at 2010 and now he is perusing Doctorate of Philosophy in Computer Science at Bharathiar University. He has 17 years of teaching experiences from various Engineering Colleges. He has published 4 papers in International journals, he has presented in 13 International Conferences & presented in 17 National Conferences in various Engineering Colleges. His area of interest includes Cloud Computing, Artificial Intelligence, Networks and Compilers.