TGA: Resource Scheduling Algorithm for Cloud Computing Environment

Theres Bemila
Assistant Professor
SAKEC, Chembur,
Information Technology

Seema Shah
Assistant Professor
VIT, Wadala,
Information Technology

Kavita Shirsat
Assistant Professor
VIT, Wadala,
Information Technology

ABSTRACT

Cloud computing is an advance computing paradigm in the Information Technology domain. It is usually used to describe the large scale distributed infrastructure, platform and software services provided by cloud service provider. Resource scheduling is one of the important tasks in cloud computing because it is involved with multiple resources like both hardware as well as software. In this paper we have described various resource scheduling algorithms used for cloud computing environment and also proposed design of new resource scheduling algorithm for cloud computing environment. So it can achieve more improvement in the resource scheduling process. This algorithm improves availability in cloud computing environment for resource scheduling as comparative with other standard algorithms. The efficiency of the user request will first be optimized and then processor executes the request. Finally it specifies fast response and execution of user request.

Keywords

Cloud Computing, TGA

1. INTRODUCTION

Cloud Computing describes a new supplement, consumption and delivery model for IT (Information Technology) services based on Internet protocols. It typically involves provisioning of dynamically scalable and often virtualized resources provided by the Internet. This may take the form of web-based tools or applications that users can access and use through a web browser as if the programs were installed locally on their own computers. The definition of Cloud Computing provided by NIST (National Institute of Standards and Technology) says that: "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [1]. It can also be expressed as: Cloud Computing is such a type of computing environment, where business owners outsource their computing needs including application services to a third party and when they need to use the computing power or employees need to use the application resources like database, emails etc they access the resources via Internet.

2. MOTIVATION

Cloud Computing is a technology which uses Internet and remote servers to maintain data and various applications. Cloud services help to improve IT capabilities without investing large amounts in new data centers or servers. Cloud Resource Scheduling plays an important role in providing various Cloud services to the end users. To achieve the full benefits of Cloud Computing, effective and efficient resource scheduling algorithm is fundamentally important. However, traditional resource scheduling algorithms encounter a number of challenges. Thus, the main purpose is to provide proper resource scheduling algorithm through effective performance (i.e. response and processing time) with the available cloud resources. During the resource scheduling in cloud environments how to make full use of resource and how to effectively select resource are important aspects. At the same time, communication delay also plays an important role in it, which not only leads to waiting between tasks but also results in much idle interval time between processing units. This paper presents the design of new cloud resource scheduling algorithm for the available cloud resources.

3. CLOUD RESOURCE SCHEDULING

Cloud Resource Scheduling is the process of deciding how to commit resources between collections of possible tasks. It decides which set of resources are available i.e. free and allocated. It is needed in case of multitasking and multiprocessing environment. Also it is the process of mapping tasks to available resources on the basis of tasks characteristics and requirements. The available resources should be utilized efficiently without affecting the service parameters of Cloud. Cloud models use virtualization technology; this technology helps in slicing a single data center or server to act as multiple machines. It depends on the hardware configuration of the data center into how many VM (Virtual Machine) they can be divided. To implement virtualization additional software is also required. This software is the system software an operating system, can be from Windows e.g. Windows Server 2008 or Hyper-V or for an open source environment like Linux or Xen. Resource scheduling is the pre requirements for increasing the Cloud performance and for utilizing the resources [2].

Cloud Resource Scheduling can be centralized or decentralized. Its several algorithms are introduced like RR (Round Robin) algorithm, with a mining improvement in the performance. The only differences with this algorithm are in their complicity. The effect of the algorithm depends on the architectural designs of the Cloud. Today Cloud Computing is a set of several data centres which are sliced into virtual servers and located at different geographical location for providing services to clients [3]. The objective of the paper is to suggest resource scheduling algorithm for such virtual servers for higher performance rate.

This paper includes the four fundamental algorithms for Cloud Resource Scheduling namely RR, ESCE (Equally Spread Current Execution Load), TLB (Throttled Load Balancing) and FCFS (First Come First Serve) algorithms. Cloud Resource Scheduling works in the manner to decide which VM is in ready state while which VM will go on hold state. It helps in reducing the BW (Band Width) usage which results in decreasing the cost of machine and maximizing the services offered by Cloud.

4. RELATED WORK

Now the paper describes about four cloud resource scheduling algorithms which are RR, ESCE, TLB and FCFS in detail.

4.1 Round Robin

It uses the time slicing mechanism. Using this algorithm, the scheduler allocates one VM to a node in a cyclic manner. The scheduler starts with a node and moves on to the next node, after a VM is assigned to that node. This is repeated until all the nodes have been allocated at least one VM and then the scheduler returns to the first node again. Hence, in this case, the scheduler does not wait for the exhaustion of the resources of a node before moving on to the next. The name of the algorithm suggests that it works in the round manner where each node is allotted with a time slice and has to wait for their turn. The time is divided and interval is allotted to each node. Each node is allotted with a time slice in which they have to perform their task. Open source simulation software knows as Cloud Analyst, this algorithm is the default algorithm used in the simulation. This algorithm simply allots the job in round robin fashion which doesn't consider the load on different machines.

4.2 Equally Spread Current Execution

This algorithm requires a resource scheduler which monitors the jobs which are asked for execution. The task of resource scheduler is to queue up the jobs and hand over them to different VMs. The scheduler looks over the queue frequently for new jobs and then allots them to the list of free virtual server. The scheduler also maintains the list of task allotted to virtual servers, which helps them to identify that which VMs are free and need to be allotted with new jobs. The experimental work for this algorithm is performed using the Cloud Analyst simulation. The name suggests about this algorithm that it work on equally spreading the execution load on different VM.

4.3 Throttled Load Balancing

It works by finding the appropriate VM for assigning a particular job. The job manager is having a list of all VMs, using indexed list, it allot the desire job to the appropriate machine. If the job is well suited for a particular machine than that job is, assign to the appropriate machine. If no VMs are available to accept jobs then the job manager waits for the client request and takes the job in queue for fast processing[4][5][6].

4.4 First Come First Serve

It is useful for parallel processing and is aiming at the resource with the smallest waiting queue time and is selected for the incoming task. The queue is maintained to process the request task for each node which comes from the ready queue. Throughput is low because of long process hold the resources. Starvation is possible since waiting time for long process may be longer. The complicity of this algorithm is less compared to the other three algorithms. This can be called as Opportunistic Load Balancing (OLB). Using this algorithm, the scheduler prioritizes the jobs in the queue according to a number of factors and then orders the jobs into a highest priority first or priority FIFO/sorted list. The scheduler starts with a node and moves on to the next node, after a VM is assigned to that node. This is repeated until the nodes have been allocated to VM [7].

After the detailed study of various cloud resource scheduling algorithms, there is great need to have proper cloud resource scheduling algorithm in order to fulfil the promising cloud services. Now present the design of new proposed cloud resource scheduling algorithm i.e. TGA (Tailored Genetic Algorithm) which is tailored to fulfilled the proper utilization of cloud resource.

5. DESIGN OF PROPOSED TGA

Now present the design of proposed new Cloud resource scheduling algorithm i.e. TGA which consist of different designing steps. They are fitness function which will decide the tendency of the evolution, extracting record which will fetch the value of allocation and last is crossover which will find out best value for allocation. They are presented below.

The most important step of this algorithm is to find the fitness function used in TGA. To use TGA in the combinational optimization problem the most important factors are chromosome representation, fitness function and genetic operators i.e. crossover and mutation. This paper presents the design of new crossover technique because of the unique suggestion value that it specifies as the resultant. This optimal scheduling technique reallocates resource for queuing or instant VM requests each time either the resource statutes change or new VM requests come. Hence it is also an automated, immediate and instant scheduling technique.

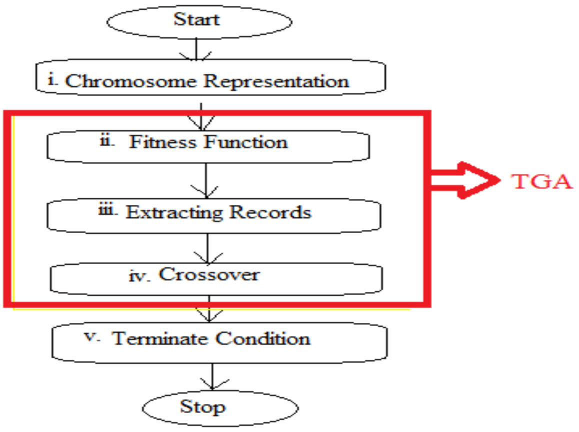

Figure 1: Flowchart of Tailored Genetic Algorithm

Following are the steps involved in TGA. They are as follows as:

i. Chromosome Representation:

Each VM request comes through a unique VmId i.e. Virtual Machine Identifier. Since VmStatesList gives the total number of available or created VmId by the user request based on their states. There are basically two states for VM, one is AVAILABLE and other is BUSY. These states value can be fetched through the CloudSim Event Listener class. A direct representation of solutions, with a chromosome encodes a scheduler. So for N VM Requests to be allocated on M computing nodes i.e. Data Center (DC), the chromosome will hold N genes representing the VM Requests which will be scheduled in sequence. A single chromosome or a single user request can be represented as follows as:

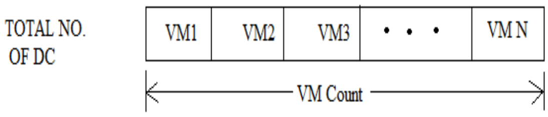

Figure 2: Chromosome Representation in Cloud Environment

As in the above Figure2, VM count is equal to VmStatesList.size (). Consider N VM Requests need to be allocated. And suppose there are M idle or partly idle computing nodes i.e. DC available in a Cloud. First number the computing nodes with integer value. For M computing nodes given each of them a unique integer arrange from 0 to M-1, and mark each of the computing node's (i.e. CPU cores, memory size and hard disc size). Second, the TGA algorithm holds the VM Requests sequentially in a list and records their content. Then, the TGA will produce a chromosome using N integer genes obtained from VmStatesList variable, using the variable along with size() function i.e. VmStatesList.size() helps to create the total number of VM Request need to be allocated . And value of the genes is limited from 0 to M-1. And then set an appropriate population size and max evolutions for the evolution. TGA will generate all possible sequences of these nodes like DC1, DC2, DC3, etc. Thus, the decoding is very simple. The decoded integer is the sequence number of the computing node. And the scheduler will allocate certain physical resource for the VM Requests from left to right by value of the integer genes. Since, it launches multiple instances on one node. A valid sequence must have exactly one occurrence of each node. And each job will be allocated only once.

ii. Fitness Function

This function is an important step of any GA. It will decide the tendency of the evolution. In this proposed algorithm, consider the Dividend Policy in Economics as a reference. There are two parts in a Dividend Policy; they are capital gains and the dividend yield. The capital gains represent the fitness value of each gene in a single chromosome representation while the dividend yield is an overall condition of the gene duplication and they are denoted as follows as in Formula (1).

$$K = \text{capital gains} + \text{dividend yield} \qquad (1)$$

Formula (1) is a general idea of Dividend Policy in Economics. The situation brings in a max fitness value and takes a deduction method. Hence, it will keep none negative fitness value. Formula (2) is a modified Dividend Policy. The main idea is that given a total value of the whole genes. And divide the value to each part of the corresponding dynamically. A sensible max value gives as the most fitness one. For not calculate a negative value, use a subtraction method to deduct the inappropriate parts. And the deduction part will not exceed the max value in reasonable situations.

$$F = Max\_Fitness - K \qquad (2)$$

In Formula (2), F is the fitness value of a chromosome after Max_Fitness minus K, K is a certain value of the resource that current chromosome wasted using the Dividend Policy.

$$K = \sum_{i=0}^{n-1} \sum_{j=0}^{2} C_{ij} + D \qquad (3)$$

$$C_{ij} = \begin{cases} = \dfrac{Max\_Fitness}{3 * vmStatesList.size\ ()} + 1 - X \\ \text{where,} \quad X >= 1 \qquad \text{OR} \\ = 1 - X \quad \text{where,} \quad X < 1 \end{cases} \qquad (4)$$

And

$$X = \frac{currVm}{vmStatesList.size\ ()}$$

$$D = \frac{Max\_Fitness}{vmStatesList.size\ ()} * \ Dups \qquad (5)$$

For K in Formula (3), it is composite of capital gains and dividend yield. From Formula (4), the value of C_{ij} is depending on the result of X and the more situations fulfill the VM Requests the less it will be deducted. Where X, consist of currVm i.e. Current VM request for VM out of the total available list. Each currVm consist of DC type, memory size, Bandwidth size and image size. While vmStatesList.size() actually provides available free DC types, free memory size, free Bandwidth and free image size. Thus, in that way a fulfill target node will be encouraged while an unfit node will be prepare out probably.

In Formula (5), D represent duplicated corresponds in a generation, in which Dups is initialize to 1 because every time simulation begin duplicate value are never created and assumption is made that even if configuration are same there are no duplicate and then calculate how many VM Request should not be appeared.

iii. Extracting Records

The Cloud Analyst software is provided with simulation files if load an existing configuration for simulation. After loading .sim file various configurations like Data Center Configuration, Virtual Machine Allocation detail, User Baser requesting information and network characteristics are fetched. The Cloud Analyst software gives output as PDF (Portable Document Format) file. From the .sim file VmId i.e. Virtual Machine Identifier value is extracted and it is used for performing various operations on it namely fitness evaluation and crossover.

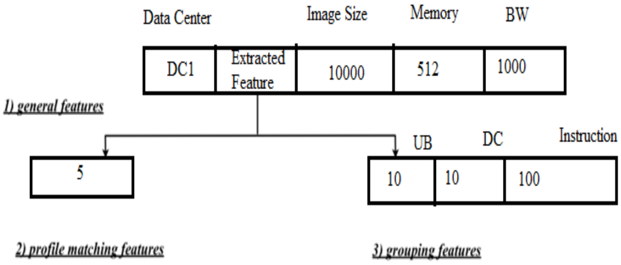

Figure 3: Extracting Records using TGA for Cloud resource scheduling

From these extracted value, can identify whether the respective collection are allocated i.e. BUSY or free for allocation i.e. AVAILABLE [8] [9].

iv. Crossover

The evolutionary process of a GA is a highly simplified and stylized simulation of the biological version. It starts from a population of individuals randomly generated according to some probability distribution, usually uniform and updates this population in steps called generations. Each generation, multiple individuals are randomly selected from the current population based upon some application of fitness, bred using crossover, and modified through mutation to form a new population. The following are phases of TGA are as follows:

- Selection phase: Chromosome features are extracted using Cloud Analyst software. The selection used in this paper, a threshold is set for VmStatesList which are rated more than the threshold value is selected. Those records which fall below threshold value are not selected and are ignored. The selected ones form the initial population for TGA, where these records value are used in the next phase of this paper.

- Crossover phase: The BLX (Blend Crossover) or alpha crossover technique is used since extracted features are real numbers. Hence crossover is performed with this algorithm resulting in new generation. The selected individuals are given to the below BLX- alpha crossover technique. This is used to generate new off springs after the crossover step.

  Crossover Steps:

```
I. Select two parents X (t) and Y (t) from a parent pool.
II. Create two offspring X (t+1) and Y (t+1) as follows:
III. For i = 1 to n do
     di=|xi (t)-yi (t)|
IV. Choose a uniform random real number u from interval <min (xi (t), yi (t))-adi, max (xi (t), yi (t)) +adi>
V. xi (t+1) =u
VI. Choose a uniform random real number u from interval <min (xi (t), yi (t))-adi, max (xi (t), yi (t)) +adi>
VII. yi (t+1) =u
VIII. End do
Where, a – positive real parameter
```

- Matching phase- This phase finds the similar items stored in VmStatesList to the newly generated CurrVm features. Once similarity is found those items are recommended for the allocation. This phase uses Euclidean distance between two offspring and distance between each feature of the two offspring is calculated, resulting value is used to match the records stored in the VmStatesList. Those records are compared with the resulting value which the user has given highest request for allocation [10] [11].

Euclidean Formula

$$dij = \sqrt{\sum_{k-1}^{n} (x_{ik} - x_{jk})^{2}}$$

Where i and j are two values and k is the identifier of virtual machine.

v. Terminate Condition

Using this TGA algorithm, tried finding the VmId i.e. Virtual Machine Identifier those which are available and used. Then fitness function is applied on it. This is then followed by passing parameter to crossover function; the output of crossover gives the best individual which is needed to be allocated. Then VM is allocated with that respective value and through this procedure get next available VM with the unique value after performing different steps it is returned. Now present the working of TGA in the next section.

6. EXPERIMENTAL WORK

In the proposed Cloud resource scheduling algorithm, it initially extracts the unique features of each item using Cloud Simulation software. As inputs to TGA, the individuals are composed of the extracted features from the simulation configuration / .sim file. Users request to the system will be evaluated by calculating the fitness of resource. Next, TGA operates on the evaluated value and then discovers the most appropriate value of virtual machine to be allocated more rapidly by assigning proper identifier for virtual machine. TGA proves to provide a stable and faster response time as well as processing time with respect to simulation result by showing stable and good performances in different scenario.

7. CONCLUSION

In this paper described various Cloud resource scheduling algorithms. Existing resource scheduling algorithms gives high throughput and cost effective but they do not consider proper utilization of available resource i.e. availability. So proposed design of new algorithm i.e. TGA that improves availability in Cloud computing environment for resource scheduling as comparative with other standard algorithms. The efficiency of the user request will first be optimized and then processor executes the request. Thus, it provides faster response and processing of various user requests (UBs) to the cloud data centers (DC).

Cloud Computing is a wide concept and resource scheduling plays a very important role in case of Cloud Computing. This paper discussed a new resource scheduling algorithm i.e. TGA that can be applied to Clouds, but still there are other algorithms that can be applied to balance the load in Clouds, improve utilization of Cloud resource in terms of reliability and availability. The performance of those algorithms can also be increased by varying different parameters like cost, throughput, QoS (Quality of Service), energy consumption etc. Heuristic algorithms can be also implemented on the Cloud simulator for Cloud Computing environment and results can be generated. Then those algorithms performance and behavior for different set of parameter can be evaluated and tested. While those algorithms will have an increased level of the complexity associated with it.

8. REFERENCES

[1] http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

[2] Anthony T. Velte, Toby J. Velte, Robert Elsenpeter," Cloud Computing: A Practical Approach", The McGraw-Hill Companies (2010).

[3] Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanzadeh and Christopher Mcdermid," Availability and Load Balancing in Cloud Computing", IACSIT 2011.

[4] Jasmin James, Dr. Bhupendra Verma "Efficient VM load balancing algorithm for a cloud computing environment" IJCSE, 2012.

[5] Brain Underdahl, Margaret Lewis and Tim mueting "Cloud computing clusters for dummies" Wiley Publication (2010).

[6] Bhathiya Wickremasinghe "Cloud Analyst: A Cloud-Sim-Based Tool For Modeling and Analysis of Large

Scale Cloud Computing Environments. MEDC Paper", Report2009.

[7] Theres Bemila, Dr. Seema Shah "Resource Scheduling for Cloud Computing Environment based on FCFS Technique", International Conference on Emerging Trends in Technology & its Applications (ICETTA),2014.

[8] Hai Zhong, Kun Tao, Xuejie Zhang, "An Approach to Optimized Resource Scheduling Algorithm for Open Source Cloud Systems", The Fifth Annual ChinaGrid Conference, IEEE, 2010.

[9] Dr. S. N. Sivanandam, Dr. S. N. Deepa, "Principles of soft computing", WILEY-INDIA, Second Edition.

[10] Xiaoyuan Zhu, Yongquan Yu, Xueyan Guo, "Genetic Algorithm based on Evolution Strategy and the Application in Data Mining", First International Workshop on Education Technology and Computer Science, pp- 849-852, 2009.

[11] D.E Goldberg and J. H Holland, "Genetic Algorithms and Machine Learning", 2006.