

Flower Pollination Based Task Scheduling With Cost Minimization Effort Using Multi Cloud Resource Provisioning

Ponni S⁽¹⁾, Hemapriya K⁽²⁾, Inkersal M⁽³⁾, Monika A⁽⁴⁾

Assistant Professor, Department of Computer Science Engineering, Velalar College of Engineering And Technology,
Thindal, Erode, India¹

U.G. Student, Department of Computer Science Engineering, Velalar College of Engineering And Technology,
Thindal, Erode, India^{2,3,4}

ABSTRACT: In the cloud condition, the work processes have been oftentimes used to display enormous scope issues in zones, for example, bioinformatics, space science, material science and number-crunching process. Such an asset acquires an assignment from the cloud suppliers that has ever-developing information and registering necessities and consequently requests an elite processing condition so as to be executed in a sensible measure of time. These work processes are ordinarily demonstrated as a lot of assignments interconnected by means of information or registering conditions. The best asset choice is finished by the examination of a few elements like anticipated that time should execute an errand by client and access limitation to assets to utilize assets. To accomplish this, a calculation is proposed for VM movement utilizing Modified Flower Pollination Algorithm. Also, this undertaking proposes an asset provisioning and booking methodology for logical work processes on Infrastructure as a Service (IaaS) and Platform as a Service (PaaS). This venture displays a calculation which intends to limit the general work process execution cost while fulfilling time constraint limitations. The principle extent of the task is utilized to break down best accessible asset in the numerous cloud condition rely on the all execution time and information move time from source to goal. The project is planned utilizing NET BEAN 6.8. The code language is Java 1.7 and MS SQL Server 2008 as back end.

KEYWORDS: Cloud Computing, Virtual Machine, Task Scheduling.

I. INTRODUCTION

Dispersed frameworks are improving a slanting methodology in computational and logical based serious applications. Elite on distributed computing is accomplished by utilize the common pool of assets like calculation, stockpiling and so forth are offer as an assistance through web get to. The fundamental difficulties of distributed computing are task booking and offsetting the heap with suitable assets. Booking of the undertaking is profoundly founded on Quality of administration parameters it incorporates cost, time, vitality, cutoff time and so forth. Parameters in Quality of administration will be shifts from one client to different clients. Some client will require savvy planning while others require effective asset based assignment execution. Data about accessibility of assets are known uniquely by the cloud dealer. The distributed computing execute the virtualization idea. A colossal number of virtual machines are promptly accessible to assign the errands. In any event, having a gigantic number of VMs a portion of the virtual machine are over used and some VMs are underutilized. Cloud scheduler has an obligation to plan the assignment to the VMs, it results in effectively used virtual machines. Cloud scheduler will distribute the errand to the accessible assets for the execution of assignment. Compelling observing of the assets help the cloud scheduler to figure how often the assets is distributed to play out a portion of the specific errand. On the off chance that the assignment isn't get executed on schedule, at that point scheduler make the undertaking moved to another VM. The thought process of planning the assignment is to advance the exhibition figuring with an appropriate streamlined portion of assets so client is fulfilled just as assets are all around used [14]. So as to upgrade the cloud execution, the mapping of the assignments to the enhanced assets will improve adequacy and efficiencies of undertaking execution and asset usage [7]. Its outcomes in tranquil and make solace to finish the undertakings with least measure of time.

II. EXISTING SYSTEM

Distributed computing is one of the prominent innovation. It executes the virtualization idea [8]. In cloud huge number of VMs are prepared for errands execution [9]. The significant difficulties in cloud server farm is to do execution of

errand with least measure of time, least expense and mapping of suitable virtual machine to specific assignment [10]. To successfully and proficiently use all the VM that outcomes with limit the inert time and to guarantee the adjusting all the heaps. The Modified Flower Pollination Algorithm builds up a calculation for VM movement. This changed Flower fertilization based calculation finds the best VM among all accessible virtual machines by utilizing the wellness work. The task execution time is discovered which is the absolute execution time of virtual machine to finish execution of the undertaking which is the sum of task processing time and task waiting time. The rundown of VMs are taken as a contribution from the server farm. In this way the enhanced asset portion is done for the VM relocation, it will be reasonable just for the single cloud specialist co-op condition. The significant difficulties we confronted is information move cost isn't considered between various cloud communities and shortage of assets is gotten even in the single uproarious supplier condition.

III. PROPOSED SYSTEM

The Modified Flower Pollination Algorithm develops an algorithm for VM migration. This modified Flower pollination based algorithm finds the best VM among all available virtual machines by using the fitness function. The task execution time is found out which is the total execution time of virtual machine to complete execution of the task which is the sum of task processing time and task waiting time. The list of VMs is taken as an input from the data center. Therefore the optimized resource allocation is carried out for the VM migration. The framework for the proposed approach is shown in Fig. 1. VM Migration

Gupta et.al, proposed Flower Pollination Algorithm based task scheduling algorithm to minimize the idle time of the virtual machines [1]. The proposed system includes the process which can be referred in multi cloud provider’s environment for consumption time and execution cost estimation for running the processes. The scheduling process is done in the basis of set of resources, number of task which are defined to that resource in the environment. To computing the result of total consumption cost and total execution time using deadline resource provisioning logic.

Shukla et. al., study importance and challenges of quality of services which affects the services of cloud computing. If the resources in cloud environment are properly scheduled, it reduced the response time, total cost of cloud resources, energy consumption and enhance the performance of overall environment [6]. It helps to assigns different options for the selection of initial resource pool .ie. For the given task, the different set of initial resource requirements is assigned. In addition, data transfer cost between data environment are also calculated so as to minimize the cost of execution in multi-cloud service provider environment.

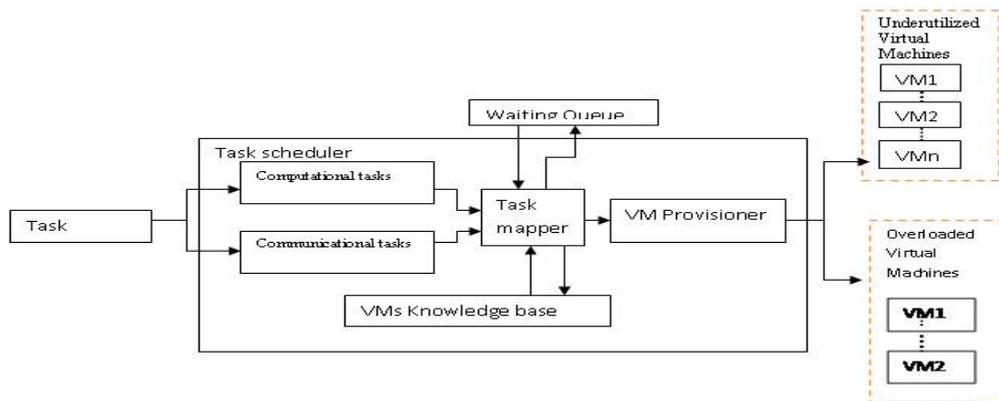


Fig 1. Proposed Cloud Framework

Task Execution Time: It is the total execution time of virtual machine to complete execution of the task which is the sum of task processing time(T_{pt}) and task waiting time(T_{wt}).

$$\text{Task waiting time}(T_{wt}) = \max_input(T)/BW \quad (1)$$

$$\text{Task processing time}(T_{pt}) = \text{task size}/MIPS(VM) \quad (2)$$

$$T_{ET} = \sum(T_{pt} + T_{wt}) \quad (3)$$

$$\text{Min_Makespan} = Sd.M_VM(T_{ET1}, T_{ET2}, \dots, T_{ETm}) \quad (4)$$



If tasks have no dependency in that case T_{wt} will be 0, otherwise it is to be calculated.

Cost: Cost of the VM for execution of the task is sum of the cost of task execution per instruction and cost of task migration from m^{th} VM to n^{th} VM and cost of the storage for execution.

$$C_p = T_{ET} * Cost_{per\ task} \tag{5}$$

$$C_{sto} = (T_{et} + T_{wt}) * Cost_{per\ storage} \tag{6}$$

$$C_{trans} = \frac{data(T_i, T_j)}{BW(m, n)} * Cost_{transfer} \tag{7}$$

$$C_{TC} = \sum_{t \in T} (C_p + C_{sto} + C_{trans}) \tag{8}$$

$$C_{total} = \sum C_{TCi} \tag{9}$$

C_{TC} is the total cost of VM for execution of tasks. C_{total} is the total cost of all virtual machines running on a data center.

Throughput: It is denoted as T_{total} i.e. number of millions of instructions by the no. of VM. Total throughput of data center is the average throughput of all virtual machines.

$$T = \frac{\sum_{t \in T} MI(t)}{TET} \tag{10}$$

$$T_{total} = \frac{\sum_{i=1}^m T}{m} \tag{11}$$

$$\text{Fitnessfunction: } C_{total} * Min_Makespan + T_{total} \tag{12}$$

a) Flower Pollination Algorithm Based VM Migration

Input: VMs from the data center;

Expected Result: VM migration with efficiency;

1. For all VMs {VM1, Vm2, ..., VMn} ;
2. Find the minimum makespan time of available VM using eqs(1-4)
3. Find the cost of available VMs using eqs(5-9);
4. Find the throughput of the available VMs using eqs(10-11);
5. Find the fitness function by eq. 12.
6. IF(Fitness_value < THRESHOLD value)
7. Implement flower pollination algorithm[1][15] for VM migration ;
8. IF (THRESHOLD value > Fitness_value) then
9. Break;
10. Process continues until it reaches NULL.
11. Break;
12. End;

The proposed algorithm use Flower pollination used to sole the optimized resource allocation using VM migration [1]. The proposed algorithm is used to find the best VM among all available VMs.

b) Components of Proposed Framework

1. Cloud Providers: This module is used to add the cloud provider details to the database table. The cloud provider id and the cloud provider name are added to the table. All the record details can be viewed using the Grid View control in a form. The details are store to 'CloudProviders'. The resource details must include which cloud provider id it belongs to.



2. Resources: This module is used to add the resource details to the database table. The resource id and the resource name and cloud provider id are added to the table. All the record details can be viewed using the Grid View control in a form. The details are store ‘Resources’ table. The cloud provider ids are fetched from the ‘CloudProviders’ table and any one id is selected as resource type for this record.

3. Process: This module is used to add the process details to the database table. The process id and the process name are added to the table. All the record details can be viewed using the Grid View control in a form. The details are store on the ‘Processes’ table. The details regarding which process use which resource is added later. The task details contain which process it belongs to.

4. Assign Process/Resource: This module is used to add the process/resource details to the database table. The process_resource id (used as primary key), process id and the resource id are added to the table. All the record details can be viewed using the Grid View control in a form. The ‘Process_Resource’ table is used to store the records. The details regarding which process use which resource is assigned here.

5. Tasks: This module is used to add the task details to the database table. The task id and the task name and process id, resource id and time taken in that resource are added to the table. All the record details can be viewed using the Grid View control in a form. The ‘Task’ table is used to store the records. The process ids are fetched from the ‘Processes’ table and any one id is selected as process id for this record. The relationship between process id and task id is one-to-many relationship.

6. Execution Time Matrix Generation: This module generates the execution time matrix in which number of resources is taken as columns and tasks are taken as rows and the time the tasks taken to complete in those resources are stored as values.

7. Transfer Time Matrix Generation: This module generates the transfer time matrix in which number of taken are taken as columns and rows (square matrix is prepared) and the time a task transfers the data to other task is stored as values. So the diagonal elements are always zero since same task has no data transfer operation.

8. Schedule Generation: Initially, the set of resources to lease R and the set of task to resource mappings M are empty and the total execution cost TEC and time TET are set to zero. After this, the algorithm estimates the execution time of each workflow task on every resource $r_i R_{initial}$. This is expressed as a matrix in which the rows represent the tasks, the columns represent the resources and the entry $ExeTime_{i,j}$ represent the time it takes to run task t_i on resource r_j . This time is calculated using Fig 2. The next step is the calculation of the data transfer time matrix. Such matrix is represented as a weighted adjacency matrix of the workflow DAG (*Directed acyclic graph*) where the entry $TransferTime_{i,j}$ contains the time it takes to transfer the output data of task t_i to task t_j . This value is taken from database and is zero whenever ij or there is no directed edge connecting t_i and t_j . An example of these matrices is shown in Figure 2 and 3 below.

$$exeTime = \begin{matrix} & r_1 & r_2 & r_3 \\ \begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \\ t_9 \end{matrix} & \begin{bmatrix} 2 & 1 & 4 \\ 4 & 3 & 6 \\ 10 & 6 & 15 \\ 7 & 4 & 12 \\ 8 & 4 & 10 \\ 3 & 2 & 7 \\ 12 & 7 & 18 \\ 9 & 5 & 20 \\ 13 & 8 & 19 \end{bmatrix} \end{matrix} \quad (a)$$

Fig 2. Matrix representation of execution time

$$transferTime = \begin{matrix} & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 & t_9 \\ \begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \\ t_9 \end{matrix} & \begin{bmatrix} 0 & 9 & 9 & 9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix} \quad (b)$$

Fig 3. Matrix representation of transfer time

IV. RESULTS AND DISCUSSIONS

Distributed systems are enhancing a trending approach in computational and scientific based intensive applications. The main challenges of cloud computing is task scheduling and balancing the load with appropriate resources. Some user will require cost effective scheduling whereas others require efficient resource based task execution. Information about availability of resources are known only by the cloud broker. The cloud computing implement the virtualization concept. An enormous number of virtual machines are readily available to allocate the tasks. Even having an enormous number of VMs some of the virtual machine are over utilized and some VMs are underutilized. Cloud scheduler has a duty to schedule the task to the VMs, it results in efficiently utilized virtual machines. The motive of scheduling the task is to enrich the performance computing with a suitable optimized allocation of resources so that user is satisfied as well as resources are well utilized. In order to improve the cloud performance, the mapping of the tasks to the optimized resources will improve effectiveness and efficiencies of task execution and resource utilization. Its results in stress free and make comfort to complete the tasks with minimum amount of time.

V. FUTURE ENHANCEMENTS

The system is very flexible and user-friendly, so the maintenance based on the changing environment and requirements can be incorporated easily. Any changes that are likely to cause failures are prevented with security and preventive measures could be taken. The coding is done in understandable and flexible method program which helps easy changing. Since MS-SQL Server and Java are very flexible tools, user can easily incorporate any modular program in the application.

REFERENCES

1. Gupta, I., et. al, "A Flower Pollination Algorithm Based Task Scheduling in Cloud Computing", International Conference on Computational Intelligence, Communications, and Business Analytics, 2017, DOI: 10.1007/978-981-10-6430-2_9, pp-97-107
2. Anbazhagi, L. Tamilselvan and Shakkeera, "QoS based dynamic task scheduling in IaaS cloud," *2014 International Conference on Recent Trends in Information Technology*, Chennai, 2014, pp. 1-8. doi: 10.1109/ICRTIT.2014.6996176.
3. Pandey, Suraj, et al. "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments." *Advanced information networking and applications (AINA), 2010 24th IEEE international conference on*. IEEE, 2010.
4. Jang, Sung Ho, et al. "The study of genetic algorithm-based task scheduling for cloud computing." *International Journal of Control and Automation* 5.4 (2012): 157-162.
5. L. Liu, M. Zhang, R. Buyya. "Deadline-constrained co evolutionary genetic algorithm for scientific workflow scheduling in cloud computing." *Concurrency and Computation: Practice and Experience* (2016).
6. S. Shukla, A. K. Gupta, S. Saxena and S. Kumar "An Evolutionary Study of Multi-Objective Workflow Scheduling in Cloud Computing." *International Journal of Computer Applications (0975-8887), Volume* (2016).
7. P.Kumar, M.Jaeyalakshmi, "Task scheduling using meta-heuristic optimization techniques in cloud environment", *International Journal of Pure and Applied Mathematics*, Volume 118 No. 18 2018, ISSN: 1314-3395.
8. Barham, P. et al. (2003), "Xen and the Art of Virtualization", *Proc. 19th ACM Symp. Operating Systems Principles (SOSP'03)*, Bolton Landing, USA, October 19-22, pp. 164-177. ACM, New York, USA.
9. Zhang, Q., Cheng, L. and Boutaba, R. (2010), "Cloud Computing : state-of-the-art and research challenges". *Internet Serv. Appl.*, Vol1, pp.7-18.
10. Fang, Y., Wang, F. and Ge, J. (2010), "A task scheduling algorithm based on load balancing in cloud computing", *Web Information Systems and Mining*, Lecture Notes in Computer Science 6318, Springer, Heidelberg, Germany. pp. 271-277.