Indian Journal of Science and Technology, *Vol 9(37), DOI: 10.17485/ijst/2016/v9i37/102059, October 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Dynamic Scheduling in Cloud Computing using Particle Swarm Optimization

P. Akilandeswari*, H. Srimathi and Deepali

Computer Science and Engineering, SRM University, Chennai 603203, Tamil Nadu, India;
akilandeswari.p@ktr.srmuniv.ac.in, srimathi.h@srmuniv.ac.in, deepalipatna93@gmail.com

Abstract

Objectives: Dynamic assignment of tasks to different VM in cloud datacenter and Load-Balancing by migration of tasks from an Overloaded VM to Candidate VM. **Methods/Statistical Analysis**: Particle Swarm Optimization (PSO) algorithm is used finding each iteration the P_BEST (Personal Best Solution) i.e. current solution is compared with the G_BEST (Global Best Solution) i.e. previous best solution and the G_BEST value is updated at each iteration for calculating the minimum execution time. The parameters being calculated are Global_BEST Solution Function, Personal_BEST Solution Function and Average Utilization for each processor. **Findings**: A Dynamic Approach for Task-Scheduling using the load-balancing technique is implemented in this paper. Two algorithms namely - Particle Swarm Optimization (PSO) and Improved PSO are used and a comparison is made between them based on number of performance parameters like Scheduling Length (Make Span), Total Execution Time and Total Transfer or Migration Time. A Utilization Graph is plotted to show this comparison which compares these algorithms based on their Cloudlet Length (Scheduling Length) and Total Execution Time. Improved PSO algorithm has lesser or minimum execution time as compared to the PSO algorithm because in the Improved PSO Algorithm two parameters are being considered namely- Cloudlet Length and MIPS (Million Instructions Per Second) which leads to maximum utilization of available resources by all VM. **Application/Improvements**: This approach is used for dynamically assigning tasks to VM and checking maximum utilization of available resources through load balancing and minimizing the overall execution time and migration time.

Keywords: Candidate VM, Cloud Computing, Cloudlet, PSO, Task Migration, Transfer Time, Virtual Machine

1. Introduction

Dynamic Task Scheduling in the cloud environment is an NP-Hard problem. The scheduling of tasks is done using the cloud platform. Dynamic Approach for Task Scheduling using the Improved PSO Allocation technique examined in a Heterogeneous computing environment. This Dynamic Allocation Policy can is applied to large datasets in real-world applications used for evaluating the overall system performance and also for deterministic execution.

In this approach, the different tasks are optimized by assigning them to various Virtual Machines in the cloud data Centre. This is done because the cloud workflow system owned by other service providers cannot directly control the cloud services. Hence aim is to decrease the overall scheduling cost and optimize the overall system performance by minimizing the total execution time for all tasks[1,2]. In any cloud data Centre there are some tasks either workflow tasks with QOS constraint and precedence relationship or Non-Workflow tasks with QOS constraints but no precedence relationship. The Virtual Machines are built on commodity machines with average processing speed and supercomputers with high processing power to meet various client requirements like business transaction workflow or scientific computation workflow.

*Author for correspondence

The task-level scheduling using the cloud platform involves VM (Virtual Machine) assignment for different tasks where the VM set has various Virtual Machines with their price and speed. First, the input for the task scheduling process needs to be assigned which involves the initial assignment of tasks to multiple VM (Virtual Machine) in the cloud data centre. However, few parameters need to be considered here before the task assignments are done like scalability of the cloud environment, the total number of tasks running parallel in the cloud data centre (there may be hundreds and thousands of tasks being executed). It is difficult (however not impossible) to design a scheduling policy for optimizing the task assignment to various VM (Virtual Machine) on a global scale. The most important parameter to be considered here is to reduce the overall implementation cost and also to reduce the total overhead (CPU execution time).

So in this approach, we will only focus on the task assignments to VM (Virtual Machine) for a particular local section of the entire data Centre so as to optimize the overall scheduling the process for the reasonable amount of applications (few hundred applications). Minimize the overall cost for different tasks; a parallel approach can be used for scheduling these tasks. Hence, the running overhead is optimized to a great extent. In this paper focus is made on dynamic task scheduling for a specific part of the entire data centre. There are two basic requirements for assignment of tasks to different Virtual Machines. 1. The entire global data centre reasonably and reasonable size for the task assignment to the VM (Virtual Machine) List. Scheduling Work Flow Applications to the cloud environment requires an efficient scheduling algorithm. For this a different scheduling approach is used in which all the concurrent tasks are further divided into some smaller sub-workflows. This will reduce the inter-task communication cost and it will also improve workflow execution performance by minimizing the total execution time for different sub-tasks[3].

The first requirement ensures that the local assignment of tasks to VM (Virtual Machine) list is reasonably partitioned so that the global task to VM List assignment is completely covered by the collection of the local tasks to VM List assignment. Similarly for the second requirement, control over the overheads involved in the task scheduling strategy is required wherein the overall size of local tasks being assigned to different VM must not be large for example: Consider the number of tasks to be below 1000 and number of Virtual Machines to be below 25.

For dealing with these issues so that both the requirements are fulfilled an integrated approach for assigning local tasks to the VM list has to be used. Such environment for assigning tasks to VM list will have a collection of various Virtual Machines and here we will be using a DAG Graph for representing workflow task and also their correlated non-workflow task. Each node in the DAG Graph is associated with some task or applications, each of these tasks are further assigned to Virtual Machines via integrated task to VM list assignment using the cloud paradigm.

2. Problem Description

In this paper focus is made on the Task Assignment Problem (TAP). The Dynamic DAG System will be consisting of a number of heterogeneous processors (n-processors in total) where each processor will have its own memory requirements and processing resources. Hence various tasks which are executed simultaneously on separate processors will result in different execution costs. Any task while performing a particular application will make use of these available resources.

Load Balancing Approach based on Improved PSO (Particle Swarm Optimization) technique for dynamic allocation of various tasks to these Virtual Machines over the cloud platform. The load on each and every processor eventually distributed so as to maximize the overall utilization of resources, thereby minimizing the overall execution time for all these tasks.

Primary objectives here are: 1. Minimizing the overall overheads involved in the Dynamic Task Scheduling Approach, 2. To reduce the total execution for all these parallel executing processes and 3. To improve the overall performance subjected to several resource allocation constraints[4]. Using the Fitness Function as one of the fundamental parameter we will evaluate a particle. The Fitness Function can be used as one of the performance parameter for evaluating Goodness of the Task Scheduling Policy. Objective Function is also used for estimating the total execution time for a number of parallel tasks which have been allocated to different processors.

2.1. Calculations

Global_Best Solution Function may be calculated as:

Global_Best (Gn) = (1/Processor_MaxSpan)∗ rocessors_Average Resource Utilization (1).

Where (Gn) is the Global Best Solution for the processor Pn.

This Global BEST Solution function is used for evaluating the overall quality of the task assignment. Load balancing approach uses so as to ensure that all the resource utilization is maximum and that different processing applications fully use all the resources.

The overall load on this processor reduced only if all the processor and resources are used entirely to their maximum limit. The processors which have been idle for the longer period are not considered here.

Average utilization for each processor may be calculated as:

Processor_Utilization (Pn) = Processor_Completion_Time (Pn)/Processor_MaxSpan (2)

The average processor utilization is further based on the overall performance of each and every processor. Finally by dividing the sum of all the processor utilization by the total number of processing applications, the average utilization for all the processors is calculated. Here also the processors which have been idle for quite an extended period of time are not considered so as to optimize the overall average processor utilization.

Personal_Best Solution Function may be calculated as:

Personal_Best (Px) = Max{Sum_Of_All (Global_Best Solution(Gn))} (3)

Where n is the total number of processors.

This objective function is used to calculate the average of the total execution time for different tasks or applications allocated to various processing elements. The objective function is nothing but the maximization of the above-mentioned Equation (2).

3. Problem Objective

A job or workflow application W = (T, E) can then be represented in form of DAG (Directed Acyclic Graph) where T = {T1, T2, T3,…,Tk} is the set of tasks and E = set of directed edges. Edge E of the form (Tm, Tn) represents a data dependency between Tm and Tn, where Tm is the predecessor node and Tn is the successor node.

The problem objective can be given as:

- Designing a schedule to minimize the cost function such that makespan is minimal.
- Resource utilization is expected to be maximized.
- Total Completion Time is supposed to keep to a minimum.

4. Work Flow Scheduling in Cloud Workflow Systems

In the following section brief discussion about the core architecture diagram for the cloud work flow system has been given. Also focus has been made on the various performance parameters which need to be considered while applying different Work Flow Scheduling Strategies in the cloud work flow systems.

Architecture for Cloud Workflow Systems:

In this basic architecture for the cloud workflow systems, the individual tasks which have to be dynamically scheduled are assigned to different VM belonging to a particular data center. Cloud workflow architecture is shown in the Figure 1. Based on the available resources and bandwidth of the VM overloaded VM needs to be mentioned. If any VM is overloaded the tasks are migrated from the overloaded VM to the candidate VM. Group of tasks can also be relocated in case of an overloading scenario. Particle Swarm Optimization (PSO) algorithm is used for calculating the total execution time for each task being scheduled dynamically. Various performance metrics which has to be considered is mentioned below:

5. Existing System

In cloud computing, virtualization technology enables VM migration to balance load in the data centers. Migration is done to manage the resources dynamically[5,6–8]. Live migration means the migration of a VM from source physical machine to the destination when the Virtual Machine is powered up. Virtual Machine migration should occur in such a way that it should minimize both total migration time and down time. Downtime is the time for which service is not available. Downtime is transparent to the

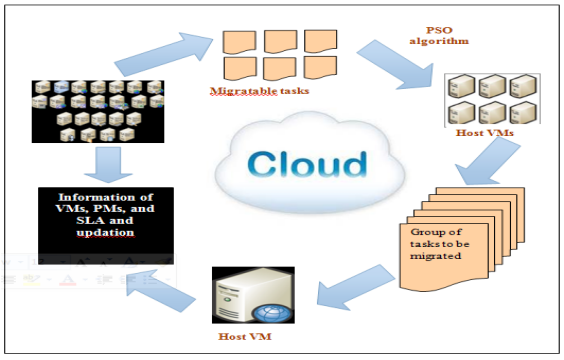

Figure 1. Basic architecture for cloud-environment.

users. Total migration time is the time from when the migration is started to when the source Virtual Machine is discarded. With Virtual Machine migration, on the other hand, the original host may be decommissioned once the migration has completed. This is particularly valuable when migration is occurring to allow maintenance of the original host.

- Issues in existing system:
  - It is costly and time-consuming.
  - The last activity information of the user is lost.
  - All the available resources are not utilized properly.
  - Total completion time is not minimum.

6. Proposed System

TBSLB method using Particle Swarm Optimization (TBSLB-PSO) develops a multi-objective optimization model for migrating tasks from overloaded VMs to minimize task execution time and task transfer time by applying the PSO algorithm. It develops a multi-objective task scheduling optimization model for migrating tasks from overloaded VMs that minimizes both task execution and transfer time and developing an MOPSO-based algorithm to solve this proposed optimization model. In addition, the TBSLB conceptual model contains a Central Task Scheduler (CTS) to transfer tasks from an overloaded VM to a new similar VM by applying the information on the blackboard. Propose an optimization model for migrating these extra tasks to the new homogeneous VMs to minimize task execution and task transfer time. TBSLB-PSO model overcomes this task migration optimization problem, using consideration of bandwidth as a variable to minimize the task transfer time for data-intensive applications. Also, it considers the properties of the new host VMs and PMs (memory, hard disc, the number of CPUs, etc.) to enhance performance utilization for computing intensive applications. It employs the information of the TBSLB blackboard as the input data for the proposed task migration model to find an appropriate host VM for the tasks. Also, it uses the PSO algorithm to solve the proposed optimization problem. Also, migration task grouping reduces the transfer time between overloaded VM and host VM. The monitoring work flow is shown in Figure 2.

- Advantages of Proposed System:
  - It is less costly and less time-consuming.

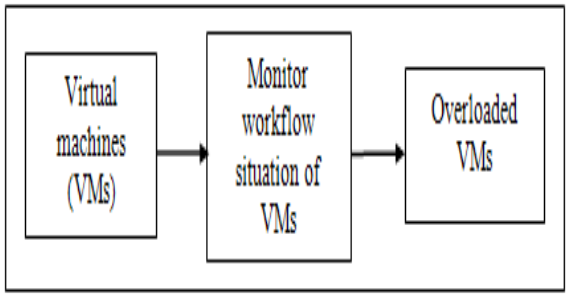

Figure 2. Monitoring VM workflow.

  - The last activity information of a user is not lost.
  - All the available resources are utilized properly.
  - Total completion time is minimum.

7. Particle Swarm Optimization (PSO) Algorithm

The PSO technique is used mainly for finding solutions for continuous optimization problems which do not have any prior information about them. To solve the task scheduling problems, we will be using a Dynamic version of PSO namely DPSO (Dynamic or Discrete PSO) or by defining the velocity and position of each particle and also determining their motion equation and associated operational rules for every discrete variable associated with each particle. Dynamic PSO using central task scheduler is shown in Figure 3. Consider that the given cloud work flow system has m some tasks and n number of resources in total.

The exact position for the particle-i is calculated as:

Xi = (xi1,x12,…,xij,…,xim) for 1<j<m and 1<xij<n.

Also each particle is associated with some velocity value Vi for each dimension as:

Vi = (vi1,vi2,…,vij,…,vim) for 1<j<m and for all vij lying in the range of {-1,0,1}.

Each particle is also associated with some local memory which stores the value for its previous best position i.e. Personal_Best Solution. Some iterations are used for finding out the best optimal i.e. least execution time for the task scheduling. Then the Global_Best Solution i.e. Global Best value is used as one of the performance parameters to which the previous results are matched in each of the iterations. Each of the previous P_Best values for a particle is then also matched with global best value i.e. G_Best and finally, the best value is selected.

Algorithm: Dynamic Task Scheduling using Particle Swarm Optimization Technique.

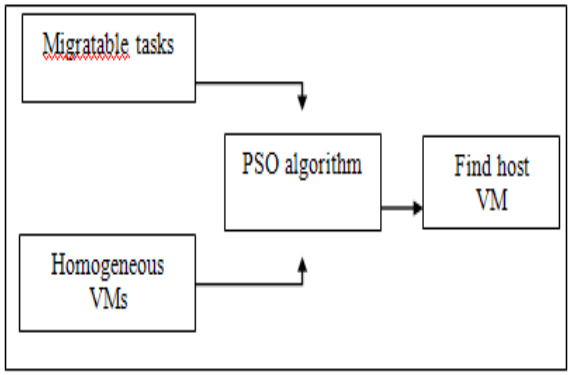

Figure 3. PSO using CTS (Central Task Scheduler).

Tasks Assigned {Tk,Total Cost_Involvedi, Time_Taken,DAG{Tk,Tk+1,Tk+2,….Tn}};
Dynamically Scheduled Tasks using the PSO Technique
Pop_Size = Population Size;
Maximum (I) = Maximum number of Iterations (I).
Minimum (I) = Minimum number of Iterations (I)
Personal_Best = Personal_Best Solution as compared with the previous iterations solution
Global_Best = Global_Best Solution as compared with the previous iterations Personal_Best Solution.
For k = 1 to Pop_Size
Generate better solution B1;
Assign this B1 to Personal_Best Solution;
Then Find the Global_Best Solution;
Calculate Learning Probability P1 and P2;
While {Iteration<Maximum(I)} and {Iteration < Minimum(I) or Personal_Best Solution<0.04 or Global_Best Solution<0.04 }
{
For k = 1 to Pop_Size
Generate some Random Number R1 for each iteration;
Calculate Vk using B1 and Personal_Best Solution;
Calculate Vx using B1 and Global_Best Solution;
Calculate V_Total using Vx and Vk;
Calculate the new position for the particle Pk using the previous values V and X;
Update (Personal_Best Solution);
Update (Global_Best Solution);
Return (Solution1, Solution_Set 1);
}
Most Optimal_Sol = COMPARE (Solution_Set1, User_Req);
Deploy (Solution).
Task migration using central task scheduler is shown in the Figure 4.

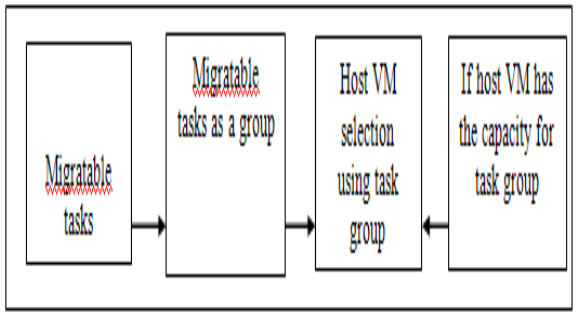

Figure 4. Task migration using CTS.

8. Module-Description and Evaluation

A static Round-Robin Scheduling Algorithm and Priority-Based List Scheduling Algorithm for the Task-Scheduling process have been used. The user will input the number of processes. For each process, some Burst Time and a particular Time Slice is allocated by the user. Using these values the Average Waiting Time and Turnaround Times are calculated for all the processes. A comparative study between The Dynamic Approach for Task Scheduling and the static approaches namely Round-Robin Scheduling Algorithm and the Priority-Based Task scheduling algorithms are done here.

The final task scheduling results are then finally evaluated using the CloudSim Simulation Tool Version 3.0.3. This Dynamic Task-Scheduling environment is comprised of some modules namely:

- Dynamic Task-Scheduling Module.
- Performance-Measure Module.
- Module for Comparative Study of Algorithms.

Performance Measures for Cloud Environment:

- Scheduling Length (Make Span):
  SL = max {FT (Ti,P)}

Calculating the Finish-Time of the exit task.

- Scheduling Length Ratio:
  SLR = {SL/Critical Path}
- Cost:

  Cost = Number of Processors * Parallel Execution-Time

Finally, Performance comparisons based on the Execution-Time is done for all the algorithms namely:

- Particle-Swarm Optimization (PSO) Algorithm.

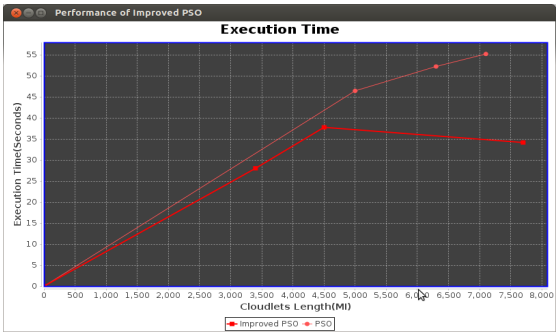

Figure 5. Utilization graph for minimum execution time.

- Improved Particle-Swarm Optimization (PSO) Algorithm.

As shown in the Figure 5, the Improved Particle-Swarm Optimization (PSO) Algorithm utilizes minimum execution time.

9. Conclusion and Future Work

With the advancements in the cloud computing environment, these Workflow System based on the cloud platform are used widely in Science as well as Business applications. The Market Oriented business models for the cloud-based systems are very different from the other basic approaches. More improved and advanced meta-heuristic approaches can be used for task scheduling algorithms in the future scenario. Hence for the handling of large-size task-level scheduling policies in the cloud platform, some various scheduling algorithms can be used. Implementation of real-world Applications can also be done for further evaluation of the system performance for these scheduling policies.

10. References

1. Guo L, Zhao S, Shen S, Jiang C. Task scheduling optimization in cloud computing based on heuristic algorithm. Journal of Networks. 2012; 7(3):547–53.
2. Mahmoodabadi MJ, Bagheri NA, Zadeh AN, Jamali. A new optimization algorithm based on a combination of Particle Swarm Optimization, convergence and divergence operators for single-objective and multi-objective problems. Engineering Optimization. 2012; 44(10):1167–86.
3. Bagheri R, Jahanshahi M. Scheduling workflow applications on the heterogeneous cloud resources. Indian Journal of Science and Technology. 2015; 8(12):1–8.
4. Komarasamy D, Muthuswamy V. A novel approach for dynamic load balancing with effective bin packing and VM reconfiguration in cloud. Indian Journal of Science and Technology. 2016; 9(11):1–6.
5. Jain N, Menache I, Naor S, Shepherd FB, Naor, J. Topology-aware VM migration in bandwidth oversubscribed datacenter networks. Springer. 2012; 7382:586–97.
6. Hai J, Gao W, Wu S, Shi X, Wu Z, Zhou F. Optimizing the live migration of Virtual Machine by CPU scheduling. Journal of Network and Computer Applications. 2011; 34(4):1088–96.
7. Jun C, Xiaowei C. IPv6 Virtual Machine live migration framework for cloud computing. Energy Procedia. 2011; 13:5753–7.
8. Sapuntzakis CP, Chandra R, Pfaff B, Chow J, Lam MS, Rosenblum M. Optimizing the migration of Virtual Computers. ACM SIGOPS Operating Systems Review. 2002; 36:377–90.