IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-5, Issue-6 E-ISSN: 2347-2693

Task Scheduling and Resource Optimization in Cloud Computing Using Deadline-Aware Particle Swarm Technique

Shruti[1*], Meenakshi Sharma[2]

[1*]Shruti, Research Scholar, GRIMT, Kurukshetra University, Haryana, India
[2]Meenakshi Sharma, Assistant Professor, GRIMT, Kurukshetra University, Haryana, India

*Corresponding Author: shrutigoel63@gmail.com

Available online at: www.ijcseonline.org

Received: 25/May/2017, Revised: 02/Jun/2017, Accepted: 20/Jun/2017, Published: 30/Jun/2017

Abstract- Cloud computing is defined as that type of computing which shows the development of potential, grid and parallel computing. It is the fastest new paradigm for delivery of services via internet. In this, the client can access software resources and valuable information over a network. It is the internet based computing in which resources are accessed via internet. In practice, the cloud computing faces the number of challenges like reliability, portability and shared access etc. Moreover, cloud computing faces the large quantity of cloud users, their tasks and data. Hence, to schedule the tasks efficiently, scheduling is done. In this paper, a Deadline Aware Particle Swarm Optimization (DAPSO) Algorithm is used which provides efficient and better results. Due to its fast convergence property, it is much better than Particle Swarm Optimization (PSO) algorithm. It is used to optimize the task scheduling algorithm which results in better performance and profit.

Keywords- Cloud computing, scheduling, Task Scheduling Algorithms, Particle Swarm Optimization (PSO), Task scheduling, Scheduling Types, Deadline Aware Particle Swarm Optimization (DAPSO)

I. INTRODUCTION

Cloud computing is emerging as an additional and most adopted access in the field of computer science. Cloud computing provides sharing of resources and data to computer and other devices on request. Due to this, it is also called on-demand computing. It provide users with various capabilities to store and process the data. It provides three service models which are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Cloud services allow users to manage and access hardware and software which in turn managed by the third party. Various examples of cloud services are online file storage, web mail, online business applications and social networking sites etc. The **cloud computing architecture** is the configuration of the system that involves local and cloud resources, services, middleware and software, cloud clients and cloud storages, networks, geo-location. It is based on the needs of the end-user and describes how all these components are arranged and related. It comprises of many cloud components which are loosely coupled. It can be categorized as front end and back end.

The front end refers to the client part of cloud computing system. It consists of interfaces and applications that are required to access the cloud computing platforms. Its example includes Web Browser. The back End refers to the cloud itself. It consists of all the resources required to provide cloud computing services. It comprises of huge data storage, virtual machines, security mechanism, services, deployment models, servers, etc.

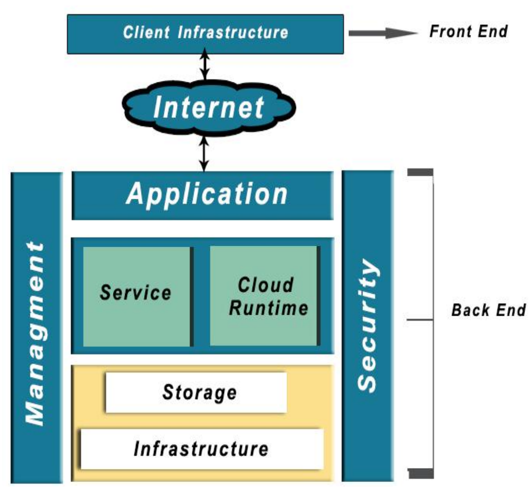

Figure 1: Architecture of Cloud Computing

Task Scheduling: Task scheduling is defined as the mapping of user tasks to virtual machine which is used to improve the performance. It comes into existence when a group of people with specified Service Level Agreement (SLA) submit their job for execution via internet. In general, scheduling is the process of mapping tasks to available resources which is based on the task's requirement. It is an essential phenomenon in the field of cloud computing. In this, the resources can be used by the user very effectively without affecting the parameters of cloud services. Various algorithms are used for task scheduling such as Simulated Annealing, Genetic Algorithm and Heuristic Algorithm etc. Among all, the Deadline Aware Particle Swarm Optimization (DAPSO) algorithm proves better for a distributed environment. In general, scheduling process in a cloud can be studied in three stages. These are –

- **Resource discovering and filtering:** In this, the resources which are present in the network system are discover by the Datacenter Broker and collects all relevant information.
- **Resource Selection:** This is the deciding stage, as, in this, the resources are selected based on the parameters of resource and task.
- **Task Submission:** In this, the task is submitted finally to the selected resource.

Types of Scheduling: The scheduling process can be classified into following categories:

- **Heuristic Scheduling Algorithm:** In this scheduling, the jobs are queued and collected into a particular set. It is used to optimize more accurate result. Its examples are First Come First Serve Scheduling algorithm (FCFS) and Round Robin Scheduling Algorithm (RR).
- **On-line Mode Scheduling:** In this type of scheduling, jobs are executed in an efficient manner. Since the cloud is heterogeneous system, therefore, the speed of each processor varies accordingly.
- **User Level Scheduling:** This scheduling consists of market based and auction based scheduling. Its examples includes Priority Based Scheduling, FIFO Scheduling and Non-Preemptive Scheduling.
- **Static and Dynamic Scheduling:** Static scheduling includes pre fetching of the required data and imposes minimum runtime overhead. On the other hand, Dynamic Scheduling is that scheduling in which the user does not know about the information and execution of tasks.
- **Real Time Scheduling:** It is used to increase the value of throughput and decrease the value of average response time.
- **Cloud Service Scheduling:** It has two parameters i.e. user level and system level. The user level is used to study the services of clients and user. The system level deals with the scheduling.
- **Priority Based Scheduling**: In this, the priority is assigned to each job which is introduced in the system. Multi-Criteria Decision making Model (MCDM) and Multi-Attribute Decision making Model (MADM) are the models which is used to set the priority.

II. RELATED WORK

Assigning the task is one of the core research area in the field of cloud computing. By taking value of constraints such as cost, time and bandwidth etc, the task assignment selects the best suitable resource in a cloud. Many scheduling algorithms were designed so as to get the proper utilization of resources.

In [2], Zhi et al. describes the adaptive particle swarm optimization to improve the performance of PSO. In this, the learning technique is used to control the inertia weight, acceleration coefficients and other algorithm parameters which is used to provide better performance. In [7], Azadi Khalili proposed a hybrid Particle Swarm Optimization (PSO) for scheduling in cloud. This type of optimization gives better results as compared to Max Min Scheduling. When tasks increases, then the average schedule length of hybrid PSO was reduced from 4.6 to 4.8% than Max Min Scheduling whose range is 2.4 to 3.4%. In [6], Salman proposed the particle swarm optimization algorithm for task assignment and the results are then compared with other heuristic algorithms. The results shows that the algorithm is much suitable for task assignment problem.

In [4], Nuttapong Netjindra et al. provided a different framework in which the number of purchased instance, purchasing option and their scheduling were considered within an optimized process. For this, they considered Particle Swarm Optimization (PSO) technique. They used this technique for cloud provisioning cost optimization. Its mechanism was to allow the generated particles to move around and at each iteration, the position of particle would be updated. In [5], BU Yanping provided an improved Particle Swarm Optimization. He tested this algorithm against the MaxMin Heuristic and the results prove that this algorithm increased the performance of the system. Its main motive was to find the best strategy for task scheduling. In [3], Xingquan Zuo et al. established an integer programming model which is used for the allocation of resource problem of an IaaS cloud in a hybrid cloud environment. For this problem, a self-adaptive learning PSO (SLPSO) is used which is based on the scheduling approach. In SLPSO, each dimension of a particle denotes a task and particle as a whole represents the priorities of tasks. This approach gives high quality scheduling solutions.

In [8], a particle swarm optimization algorithm is used to solve the task assignment problem in grid computing. Its main aim is to minimize the completion time of tasks. Its results shows that this strategy gives better results when compared with genetic algorithm. In [9], the Particle Swarm Optimization is used to assign tasks to resources. In this, he used position rule which converts continuous optimization technique to small position. This results prove that the PSO is suitable for cloud computing. In [10], ChienHung Chen et al. studied the Deadline Constrained Map Reduce Scheduling

(DCMRS) problem in heterogeneous cloud computing systems. Its main objective was to divide a job deadline into two sub deadlines: map and reduce deadlines. Then, the sub-deadlines were used for finding appropriate slots which is used to run the task. They also presented a heuristic algorithm involving the node group technique which is used to decrease the computational time.

In [12], Himani et al. introduced a soft real time scheduling approach having cost constraints and deadline constraints. They illustrates that earlier approaches are not able to meet the deadline efficiently. Therefore, they introduced a deadline-meeting methodology which is used to schedule tasks over a cloud and reduce the number of missed deadline. The results show that the given approaches was more effective in defined parameters such as Task Profit, Task Penalty and Provider profit etc.

Dr. M. Sridhar et al. [1] introduced hybrid Particle Swarm Optimization (PSO) for scheduling in cloud. In this, the best values of parameters are determined experimentally. Then the algorithms with the number of tasks (varying from 100 to 1000) were evaluated. The results show that the algorithm achieves better resource utilization. In [11], a modified PSO algorithm is used to reduce the execution time and computational cost. The experiments are evaluated under different parameters i.e. time, speed and efficiency. The result shows that modified PSO is better than ordinary PSO.

III. PROPOSED APPROACH

The task scheduling plays an important role in cloud computing. There are lots of algorithms which provides efficient mapping of resources. The task scheduling in cloud computing is implemented by a technique named Deadline Aware Particle Swarm Optimization (DAPSO) algorithm. It is much better than Particle Swarm Optimization (PSO) algorithm. It is used to optimize the task scheduling algorithm which results in better performance and profit.

Deadline-Aware Algorithm: The main objective of this algorithm is to maximize the profit, minimize the value of MakeSpan, minimize the loss and achieve deadline efficiently. This algorithm depends upon two parameters namely deadline and profit. These parameters can easily be calculated according to task and virtual machine. After this, sorting is done to decide the order of tasks and tasks mapping with virtual machine. As this algorithm is used to achieve the deadline, therefore, it is called as **"*Deadline-Aware Particle swarm Optimization*"**. The pseudo code for Deadline-Aware algorithm is given as:

Begin
Step 1: For each Tasks Xi, set the value of Deadline DLi and million instructions 'inst'.
Step 2: For each instruction i, calculate the value of MaxUserPay MAXi
MAXi = s * inst/DLi
Step 3: Put all task Xi in Queue Q.
Step 4: Perform sorting by using the deadline value DLi.
Step 5: If two tasks Xi and Xj have same value of DLi, then sorting is done according to highest value of MAXi.
Step 6: Using particle swarm optimization, mapped Xi with virtual machine.
Step 7: Using scheduling policy, schedule the tasks.
Step 8: Calculate the value of NetProfit NPi of tasks Xi which depends upon lateness Z and is given as:
if Z<0 NPi = MAXi – CostPerSec * ActualCPUTime
Else
NPi = MAXi * (1-0.01*Z) – CostPerSec*ActualCPUTime
return NPi;
End

Particle Swarm Optimization: The particle swarm optimization is one of optimization technique which is used in task scheduling. It is the variation of swarm intelligence. PSO is an algorithm given by Kennedy and Eberhart in 1995. Social behavior of organisms such as flocking of bird motivated them to look into the effect of collaboration of species onto achieving their goals as a group. The position of particles in a solution space represents a solution for the problem. Each particle has its own velocity to update its position. Each particle remembers the personal best value which is the best solution of a particle. Among the personal best from all the particles, the best one is considered as the global best. In each iteration, the velocity of particle is updated. The key terms used in PSO are:

(i) particle (individual, agent): each individual in the swarm.
(ii) position/location: coordinates of the particle which represents a solution to the problem.
(iii) swarm: collection (population) of particles.
(iv) fitness: interface between the physical problem and the optimization problem. It is a number which represents the goodness of a given solution.
(v) generation: each iteration of the optimization procedure by using the PSO.
(vi) pbest (personal best): the position in parameter space of the best fitness returned for a particular particle.
(vii) gbest (global best): the position in parameter space of the best fitness returned for the whole swarm.

The process of implementing the PSO is given by the following steps:
Step 1: Initialize the group of random particles.
Step 2: By updating the generations, search for optima.
Step 3: Find the value of "pbest" in which fitness value is stored.

Step 4: Find global best value “gbest” which is tracked by the particle swarm optimizer.
Step 5: Find local best value “lbest” which is obtained when any particle takes part in population.
Step 6: Among these three values, find the two best values.
Step 7: Calculate the velocity of particle.
PV[] = PV[] + f1* num() * (pbest[] - CP[]) + f2 * num() * (gbest[] - CP[])
Step 8: Update the position of particle.
CP[] = CP[] + PV[]

where,
PV= velocity of particle;
f1 and f2= learning factors
(f1=f2=2);
CP= current particle (solution);
num= a random number whose value lie between (0,1)

The pseudo-code of the above procedure is given as:

```
For each particle
    Initialize particle
End
Do
For each particle
    Calculate fitness value V
        If V is better than pbest
            Set current value as the new pbest
            value
    End
    Select the particle with best fitness value of all the
    particles as gbest
For each particle
    Calculate the velocity of particle
    Update the position of particle
End
```

At each step of time, the particle swarm optimization algorithm changes the velocity of each particle moving towards its pbest and gbest locations. Velocity is weighted by random terms, with different random numbers being generated for acceleration toward pbest and gbest locations respectively.

IV. EXPERIMENTAL SETUP

The main objective of deadline aware particle swarm optimization is to attain deadline and provides maximum profit. The following parameters used for this are described as below:

i) MakeSpan: It tells us about the amount of time required for completion of task. For this, DAPSO strategy proves better as compared to other strategies. It minimizes the value of MakeSpan so that the tasks can be finished efficiently and smoothly before the deadline occurs.

ii) User Loss: By using DAPSO, we can minimize the value of User Loss. Its value depends upon the lateness of tasks. DAPSO provides better results when compared with other algorithm.

iii) Task Penalty: Task Penalty is defined as the number of those tasks which are not completed within the deadline DL. The deadline aware particle swarm optimization technique missed negligible number of deadline. So, it proves better than ordinary PSO.

iv) Task Profit: It states the number of tasks which are completed successfully before deadline arrives. Here, we compare Task Profit with other strategy. The result shows that the DAPSO proves better than other policy as it is more efficient to finish the tasks before deadline occurs.

v) Provider Profit: The criteria which proves that DAPSO is better than other policy is Provider Profit. Its value depends upon lateness. The DAPSO gives maximum profit and provides efficient results.

V. CONCLUSION

This paper deals with the task scheduling strategy having two parameters - deadline and profit. We observed that early strategies are not sufficient to meet the certain criteria such as to maximize the profit and to minimize the value of response time. For this, we illustrate a new method having deadline as one of its parameter and this method is named as deadline aware optimization. We compare this algorithm with other policy. The results show that DAPSO algorithm gives better result and efficient performance. It gives the results within deadline and minimizes execution time. Also, it has fast convergence property which is used to optimize the tasks efficiently and profitably. When DAPSO is compared with other parameters such as Task Penalty, Provider Profit, User Loss and Task Profit etc. then we observed that its results are better than other policy.

VI. FUTURE SCOPE

In the future, more approaches and strategies are adopted so as to enhance the value of profit and deadline. For this, hybrid algorithm will be used. Also, the results and performance of DAPSO can be compared with other optimization policy which will be used for enhancing optimization strategy.

REFERENCES

[1] Dr. M. Sridhar and Dr. G. Rama Mohan Babu, R.V.R & J.C College of Engineering, Guntur, INDIA, 2015 IEEE International Advance Computing Conference (IACC).
[2] Zhi-hui Zhan, Jun Zhang, Yun Li, and Henry Chung, “Adaptive particle swarm optimization”, IEEE Transactions on System, Man, and Cybernetics, Vol. 39, No. 6, pp. 1362-1381, 2006

[3] Xingquan Zuo, Member, IEEE, Guoxiang Zhang, and Wei Tan, Member, IEEE Transactions on Automation Science & Engineering, Vol. 11, No.2, 2014

[4] Nuttapong Netjinda, Booncharoen
Sirinaovakul, Tiranee Achalakul Department of Computer Engineering King Mongkut's University of Technology Thonburi
Bangkok,

[5] BU Yanping1, 2 ZHOU Wei3 YU Jinshou1 1.Research Institute of Automation, East China University of Science and Technology, Shanghai 200237 China; 2.

[6] A. Salman, "Particle swarm optimization for task assignment Problem", Microprocessors and Microsystems, Vol. 26, No.8, pp.363–371, 2009.

[7] Azadi Khalili and S eyed Morteza, School of Electrical and Computer
Engineering Kashan University, Babamir, 2015 23rd Iranian Conference on Electrical Engineering (ICEE).

[8] L. Zhang, Y.H. Chen, R.Y Sun, S. Jing, and B. Yang, "A task scheduling algorithm based on PSO for Grid Computing", International Journal of Computational Intelligence Research, pp.37-43, 2008.

[9] Lizheng Guo, Shuguang Zhao, Shigen Shen, and Changyuan Jiang, "Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm", Journal of Networks, Vol.7, No.3, 2012.

[10] ChienHung Chen, JennWei Lin, and SyYen Kuo, Fellow, IEEE.

[11] Zahraa Tarek, Magdy Zakria and Fatma A. Omara, "PSO Optimization
Algorithm for Task Scheduling on The Cloud Computing Environment", International Journal of Computers and Technology, Vol. 13, No. 9, 2014.

[12] Himani, Harmanbir Singh Sidhu, Second International Conference on Advances in Computing and Communication Engineering 2015.