© 2016 IJSRSET | Volume 2 | Issue 1 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

A Truthful Optimized Task Scheduling in Cloud Computing with Consideration of user Satisfaction

A. Pavithra, A.Sunitha Nandhini, A. Vinod Kanna
Department of Computer Science and Engineering, Park College of Engineering and Technology, Coimbatore, Tamilnadu, India

ABSTRACT

Task provisioning and allocation in the cloud computing is a complex process in the real world environment with the satisfaction of user constraints. Truthful greedy mechanism was used in the earlier work which attempts to select and provide the suitable services to the cloud users and also increases the profit of cloud providers by fixing the payment function for the cloud users at run time. This work lacks from more computational time for computing and deciding the payment function for each service provider. In this work, Hybrid Bee-Particle Swarm Optimization (HBPSO) based resource allocation is introduced which would select the optimal service provider for the user request by satisfying the objectives called the makespan and monetary cost. This methodology can considerably reduce the computational cost.

Keywords: Resource Provisioning, Hybrid Bee-Particle Swarm Optimization, Scout bees, Recruiter bees.

I. INTRODUCTION

Cloud Computing is a type of computing in which the computing resources are shared to the users through the Internet. Users are required to pay for access services based on their usage and the level of QoS provided.

It is an Internet based computing in which the cloud can act as server, storage and application. Individual users can connect to the cloud from their own personal computer over the Internet. The cloud is seen as a single application to the users connected to it. Software programs, files and data used by the users are stored in the cloud server which can be accessed through the internet when needed. Even if the computer crashes, user data resides inside the cloud server.

II. METHODS AND MATERIAL

1.1 Organization

In Section 2 Preliminaries of the paper is described. In section 3 System model for Hybrid Bee-Particle Swarm Optimization algorithm is described.

TABLE 1
Notations used in this paper

Acronym	Definition
HBPSO	Hybrid Bee Particle Swarm Optimization
QOS	Quality Of Services
PSO	Particle Swam Optimization

1.2 Preliminaries

This section describes the terms used in this document and an overview of some basic elements of safe use in Hybrid Bee-PSO algorithm. In HBPSO, Bees algorithm is used for the resource allocation and PSO is used for task scheduling.

Resource Allocation:

It has two inputs: User byte value and Resource Capacities. The allocation function determines which users receive their requested bundles. ‘r’ is the requested resource of i. Allocation takes place according to the density function of the user. Output: x, optimal allocation.

Payment Calculation:

The PAY function has three inputs: b, C, x. The Payment rule determines the amount the winning user I must pay. Broker calculates pay based on the user density. The pay function also identifies j, who is the winning user in the absence of i. Output: P.

III. SYSTEM MODEL

3.1 Hybrid Bee-Particle Swarm Optimization Algorithm

The Hybrid BPSO algorithm integrates the advantages of Bees algorithm and PSO algorithm approaches in task scheduling and resource allocation. The Bees algorithm has Scout bees and Recruiter bees. The scout bees finds in which location the food is available. Recruiter bees command to take the food.

The PSO algorithm follows the biological behavior of a bird. Each bee is considered as a task, Swarm is the bee which has resources. N task is taken in random manner and the Swarm is created for all the bees. Then the objective function is calculated for each Swarm.

The objective function includes total cost model and total time model.

C_{total}=∑ (cost of execution) + (cost of receiving).

T_{total}=∑ (execution time) + (receiving time) + (weighting time).

Each Swarm acts as a particle, PSO algorithm selects the best particle.

3.2 Design Goals

In the proposed hybrid Bee-PSO (HBPSO), parameters of virtual environment like total processing capacity, internal memory and total cache of processor are initialized. Here tasks can be considered as bees. In HBPSO, Bees algorithm is used for the resource allocation and PSO is used for task scheduling. The tasks are sorted according to the deadline and stored in a stack such that the task with earliest deadline would come at the top. If there are n virtual machines available then n tasks are popped out from the task stack and one task is assigned to each of the virtual machine in a random manner. Fitness value at each of the site is calculated. Fitness value is the reciprocal of the total execution time of the executable tasks in the particular site. m best fit sites are selected in order to form swarm around each site. m tasks from the task stack are popped and assigned to the m best fit sites respectively. The newly assigned task along with the already existing task forms as a swarm.

Then the PSO algorithm takes place. A task in the swarm is considered as the particle of that swarm. Each particle has the execution time, Gstart and deadline. Gstart of a particle can be calculated as the difference between deadline and the execution time.
In a swarm gbest is initialized as zero. Particle with minimum Gstart is considered as the candidate particle. Then the next iteration starts. It is continued until all the particles in a swarm is either marked as executable or pushed into the stack.

Then the fitness value of a swarm is the reciprocal of the gbest. The fitness at each site is calculated. Based on their fitness points are sorted. If the fitness comes within the first m best points swarms are created by popping m task out from the task stack. Popped out m tasks are assigned to the m best sites respectively. The assigned task and unmarked tasks which is marked as executable before in the swarm together form a new swarm.

If any two or more of the best points are near one another, only one swarm with two gbest(gbest1,gbest2) and two candidate key(C1 and C2) is created for them. This condition encourages that there may be only one exploitation per promising region.

No overlapping is allowed during the creation of swarms. That is a task will not be assigned to more than one virtual machine If a swarm is not reporting any improvement in fitness then it is assumed to have reached a peak and is not considered again. The whole process is continued until the stack become empty.

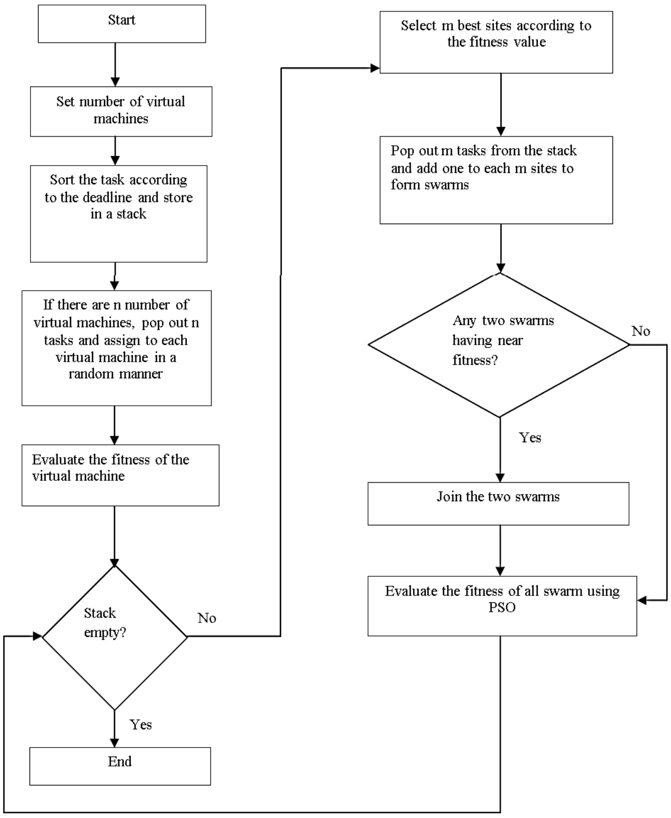
Start
Set number of virtual machines
Sort the task according to the deadline and store in a stack
If there are n number of virtual machines, pop out n tasks and assign to each virtual machine in a random manner
Evaluate the fitness of the virtual machine
Stack empty?
No
Yes
End
Select m best sites according to the fitness value
Pop out m tasks from the stack and add one to each m sites to form swarms
Any two swarms having near fitness?
No
Yes
Join the two swarms
Evaluate the fitness of all swarm using PSO

Figure 1 : Flow chart of the HBPSO algorithm for task scheduling and resource allocation

IV. CONCLUSION

The payment calculation would not be more efficient in the existing approach which needs to be processed well for the flexible accessing. The unsatisfaction of users may arise in case of wrong selection of resources. The Hybrid Bee-PSO based scheduling used in the proposed work not only confines the task execution within the deadlines but also minimizes the payments for the users.

V. REFERENCES

[1] A. Archer and E. Tardos, 2001, ‘Truthful Mechanisms for One-Parameter Agents’, Vol.42, pp. 482-491.

[2] R. Buyya, D. Abramson, and S. Venugopal, 2005, ‘The Grid Economy’, Vol.93, pp. 698-714.

[3] A. Danak and S. Mannor, 2011, ‘Efficient Bidding in Dynamic Grid Markets’, Vol. 22, pp. 1483-1496.

[4] A. Danak and S. Mannor, 2009, ‘Bidding Efficiently in Repeated Auctions with Entry and Observation Costs’, pp. 299-307.

[5] H. M. Fard, R. Prodan, G. Moser, and T. Fahringer, 2011, ‘A Bi-Criteria Truthful Mechanism for Scheduling of Workflows in Clouds’, pp. 599-605.

[6] N. Garg, D. Grosu, and V. Chaudhary, 2007, ‘Antisocial Behavior of Agents in Scheduling Mechanisms’, Vol. 37, pp. 946-954.

[7] I. Houidi, M. Mechtri, W. Louati, and D. Zeghlache, 2011, ‘Cloud Service Delivery across Multiple Cloud Platforms’, pp. 741-742.

[8] H. Izakian, A. Abraham, and B.T. Ladani, 2010, ‘An Auction Method for Resource Allocation in Computational Grids’, Vol. 26, pp. 228-235.

[9] Y. K. Kwok, K. Hwang, and S. Song, 2007, ‘Selfish grids: Gametheoretic Modeling and Nas/Psa Benchmark Evaluation’, Vol. 18, pp. 621-636.

[10] M. P. Wellman, W.E. Walsh, P.R. Wurman, and J.K. MacKie-Mason, 2001, ‘Auction Protocols for Decentralized Scheduling,’ Vol. 35, pp. 271-303.