Pak. J. Biotechnol. Vol. 13 special issue II (International Conference on Engineering and Technology Systems (ICET'16) Pp. 125 - 129 (2016)

AN EFFICIENT RESOURCE ALLOCATION STRATEGY BASED ON IMPROVED PARTICLE SWARM OPTIMIZATION (IPSO)

Naresh, T[1]

Department Sandip Institute of Technology and Research Center, Nasik. Affiliated to Savitribai Phule Pune University and Research Scholar, Computer Science and Engineering Department, at K L University

naresh1060@gmail.com

A Jaya Lakshmi[2] and Vuyyuru Krishna Reddy[3],

Department of Computer Science & Engineering, Devineni Venkata Ramana & Dr Hima Sekhar MIC College of Technology, [3]Department of Computer Science and Engineering, DST-FIST Sponsored Department, K L University, Green Fileds, Vaddeswaram - 522502.

jvallabhaneni@hotmail.com, *vkrishnareddy@kluniversity.in*

ABSTRACT

Resource allocation is a method that is deploy to allocate the accessible resource in a best way. It is the arrangement of activities and the resource needed by those activities while returns into consideration both the resource availability and the time. The proposed method discusses optimal resource discovery and dynamic resource allocation. The proposed method consists of two phases namely resource discovery and resource allocation. For resource discovery, the proposed method uses the Hierarchical Agglomerative Clustering Algorithm (HAC). Based on the HAC algorithm the proposed tree construction is generated. After that the resources are allocated by hybrid optimization algorithm. The proposed method use the Improved Particle Swarm Optimization and Cuckoo Search algorithm (IPSOCS).

Index Terms—Resource allocation, Hierarchal Agglomerative Clustering Algorithm, Particle Swarm Optimization.

I. INTRODUCTION

Resource allocation is a scheme for utilize available resources, for example human resources, mostly in the near term, to attain goals for the future. Cloud computing is internet-based computing in which large groups of remote servers are networked to allow the centralized data storage, and online access to computer services or resources. Cloud structure is assumed to be under a payment model, straight to avoid users' over - demand of their resources against their true needs [1]. Thus, a cloud replica is expected to have a scale up and down straight to manage the load variation [2]. Google furnishes a well-organized and scalable solution for working large-scale data [3]. Distributed computing suppliers convey request along the Internet, which are received from web program. The business programming and information are place away on servers at a remote area [4].

Applications can be deployed in Cloud computing in the absence of any installation of software. The users can utilize the Internet and forward messages anywhere in the world [5]. Centralized storage, memory, processing and bandwidth are greater distance systematic computing that are allowed in Cloud computing. The success of clouds has been conveyed in part by the utilization of virtualization as their unrevealed technology [6].

Cloud assists consumers to outsource computation, storage and other tasks to third party cloud providers and the resources utilized will be paid. And learn about how a cloud service provider is best multiplexing its virtual resources. The grid environment consists of several components such as schedulers, load balancer, grid broker, and portals. The schedulers are the kinds of applications that are answerable for the management of jobs, such as assigning resources required for any particular job, dividing of jobs to schedule corresponding implementation of tasks, data management [8].

Chen, *et al.,* [7] focus the difficulty of decreasing data transmission in a MapReduce cluster and the proposed method evolve a scheduling technique to better map tasks' data locality rate. When a job is assigned to the clouds, it is generally separated into several tasks. Following two questions are required to review when registering parallel processing in performing these tasks:

1) The allocation of resources to tasks;

2) Tasks are performed in what order in cloud; and

3) The planning of overheads when Virtual Machines construct, finish or handle tasks.

The scheduling of task and allocation of resource can answer these three problems.

The task scheduling application is not suitable for use in a cloud computing environment due to the small particles. The MapReduce cluster, data are distributed to separate nodes and keep in their disks. To perform a map task on a node, it requires to first have its input data available on that node [7].

Haozheng Ren *et al.,* [9] algorithm algorithm used to decrease the Communication overhead and response time of the individual job and expand the throughput of the entire system. algorithm that balances the load between the resources and expands the reliability of the grid environment. Latest internet applications are typical software performed on multi-tier architectures [10]. This algorithm consists of two-steps. In the starting step, the resources are selected based on the user time limit and fault tolerant factor of the resources [18-20]. In the next step, the load balancing algorithm is applied to decide the status of the selected resources and then the jobs are scheduled to the resources if and only if the load is balanced.

II. LITERATURE SURVEY

Resource allocation is the emerging topic in cloud computing and so many researchers already concentrated on this topic. Some of the most distinguished researches are presented here.

Prasenjit Kumar Patra *et al.,* [11] have proposed that the outcome of expansion of on demand service in computing paradigms of large scale distributed computing. It was adoptable technology as it provided integration of software and resources which are dynamically scalable. Those systems were more or less prone to failure. Fault tolerance evaluates the ability of a system to answer gracefully to an unexpected hardware or software failure. In order to achieve robustness and dependability in cloud computing, failure should be assessed and handled effectively.

Chandrashekhar S. Pawar and Rajnikant B. Wagh [12] have proposed an algorithm that is accepted Preemptable task implementation and multiple SLA variables such as network bandwidth, memory and needed CPU time. Experimental results of this algorithm showed that in a situation where resource contention is fierce, our algorithm provides superior utilization of resources.

Dorian Minarolli and Bernd Freisleben [13] have proposed the distributed version of the resource manager existing of several ANNs in which the ANN has responsible for performance of application and consumption of power of a single VM while interchanging information with other ANNs to coordinate resource allocation.

Wang Xue-yi *et al.,* [14] have proposed the cloud environment for resource allocation. First, the network structure of resource allocation mechanism was created, and then tender descriptions were given for four kinds of common resources. Second, a support vector regression (SVM) based method was acquired to change a compound resource desire into numerous isolated resource demands. Third, inner framework was established to bidding policy and tender assessment technique was designed to decide an optimal tender of cloud resource provider.

Goudarzi *et al.,* [15] have proposed a resource allocation issue which was considered to reduce the whole energy cost of cloud computing system which meet the recognized client-level SLAs in a probabilistic sense. This cloud computing system paid penalty for the percentage of a client's requests that do not meet a identified upper bound on their service time. An efficient heuristic algorithm based on convex dynamic programming and convex optimization was conferred to answer the foregoing resource allocation problem.

Doaa and Shawky [16] have proposed that the cloud resources were provisioned and released on demand. That elastic nature of resource provisioning allows for the "pay as you go" concept. Thus, it makes the major benefit of adopting cloud-based solutions. There was a need to assess the performance of the resource provisioning algorithm adopted by cloud computing platforms.

Kumar *et al.,* [17] have proposed systematic structure name called EARA (Efficient Agent based Resource Allocation) for resource allocation based on agent omputing on SaaS level in Cloud Computing. EARA Contain five different agents, each agent provided with functionality to gather details concerning all resources accessible in definite cloud deployment based on signed SLA agreement and then replied to the user with appropriate allocation or response code.

From the literature survey, it is observed that the users wish to minimize their payment when guaranteeing their service level such that their task can be finished before deadlines. Such a limit guaranteed resource allocation with reduced payment. More ever inevitable error in estimating job work load will certainly shape the problem harder.

III. OBJECTIVES

In this section discuss the objective of my research work

- To formulate a deadline driven resource allocation issue based on the cloud environment.
- To minimize user's payment in terms of their anticipated deadline.
- Analyzed the upper bound of job execution length based on the possibly inaccurate prediction
- To validate its effectiveness over a real VM facilitated cluster environment under different level of competition.

IV. METHODOLOGY

To find out the resources and assign them in a cost-effective way is the major intension of the suggested method. The suggested methodology contains 2 phases such as 1) Resource discovery and 2) Resource allocation. In the initial phase, the typical resource usage distribution for a group of nodes with related resource consumption patterns is worked out as resource bundle which can be employed to simply find a set of nodes pleasing a common obligation. Now it offers the clustering-based resource aggregation such as Modified Hierarchal Agglomerative Clustering Algorithm (MH-AC) to attain compact representation of a group of similarly behaving nodes for scalability. In the next phase for dynamic resource allocation process, the proposed method will employ bi-objective hybrid optimization algorithm based on Particle Swarm Optimization and cuckoo search algorithm. The general process of the proposed method is made cleared and shown in Fig.1. It's explained as follows:

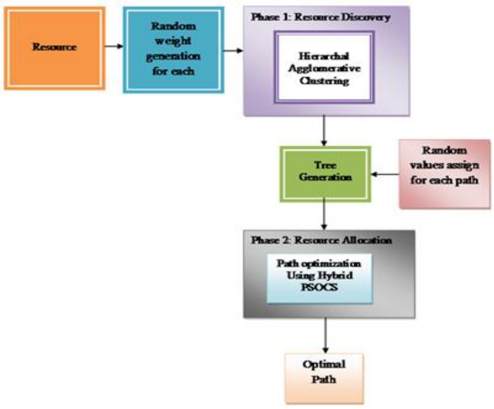

Fig.1 Hybrid optimization algorithm based on Particle Swarm Optimization and cuckoo search algorithm.

The proposed method has two phase,

1)Resource discovery

2) Resource allocation

4.1 Resource Discovery Phase

Initially, weights are produced arbitrarily for each node in the resource. The suggested method produces the tree construction based on the weights. Now reorganize hierarchical agglomerative clustering algorithm is employed for the tree construction.

4.1.1 Hierarchical Agglomerative clustering algorithm

Hierarchical clustering sequentially divides a dataset with a specified distance measure. In this sequential partition process, an algorithm erects nested partitions layer by layer through grouping objects into a tree of clusters exclusively based on the distance measure without the required to know the number of clusters in advance. There are two approaches to a tree of clusters; i.e. i) Agglomerative Hierarchical clustering algorithm or AGNES (bottom up) and ii) Divisive Hierarchical clustering algorithm or DIANA (top down). Both this algorithm is precisely overturned of each other. The top down approach presumes that all objects in a dataset are originally in a single cluster and next the cluster is recurrently separated into smaller and smaller clusters till a stopping condition is met. In difference, the bottom up approach, well recognized as agglomerative algorithm, presumes that all objects in a dataset are atomic clusters of an individual element. Next all atomic clusters combine to turn into larger. Our suggested method employs the Agglomerative Hierarchical clustering algorithm. To refine the performance of traditional agglomerative algorithm, the proposed method employ adapted Agglomerative Hierarchical clustering algorithm. Now grouping of each node is based on the minimum distance with the random weight. The specified process of adapted hierarchical agglomerative clustering algorithm with example is set below,

Step 1: Allocate each node to a detach cluster with their weights

Step 2: Begin with the disjoint clustering having level L (0) = 0 and sequence number n = 0.

Step 3: Assess all pair-wise distances between clusters

Step 4: Erect a distance matrix by means of the distance values

Step 5: Look for the pair of clusters with the shortest distance

Step 6: Increase the sequence number: n = n +1. Merge clusters (p) and (q) into one cluster to form the next cluster n. Assign the level of this cluster to L (n) = d [(p), (q)].

Step 7: Update the distance matrix, D, by removing the rows and columns corresponding to clusters (p) and (q) and adding a row and column corresponding to the recently formed cluster. The distance between the new cluster, denoted by (p, q) and old cluster(r) is described in this way: d[(r), (p, q)] = min (d [(r), (p)], d [(r), (q)]).

Step 8: Replicate till the distance matrix is decreased to a single element.

4.2 Resource Allocation Phase

Based on the hybrid optimization algorithm the resource will be assigned in this phase. To assign the resource for the task, the proposed method produce the random value for each path in the tree construction for the intention of finding the best path. Each path in the tree construction is produced a random value in the range of [0, 1]. By means of the hybrid optimization algorithm these paths are optimized. Let as reflect on a tree construction with number of path, it is revealed in Fig.2. The tree contains n number of leaf nodes and their related weights based on these each node are clustered together. To reach parent node each leaf node travel on their path. The path connected to each leaf node and their tree construction is specified beneath,

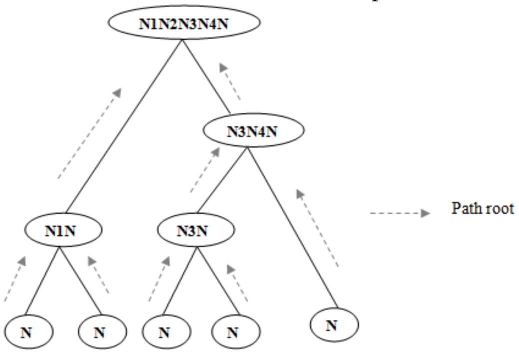

Fig 2: The tree construction with path root

Path1: (N1)-(N1N2)-(N1N2N3N4N5)
Path2: (N2)-(N1N2)-(N1N2N3N4N5)
Path3: (N3)-(N3N4)-(N3N4N5)-(N1N2N3N4N5)
Path4: (N4)-(N3N4)-(N3N4N5)-(N1N2N3N4N5)
Path5: (N5)-(N3N4N5)-(N1N2N3N4N5)

In the suggested method, it will allocate the random values for each path after finding the path for the complete leaf node. Our tree construction contains five paths, each path allocating the random values in the range of zero to one. Next the paths are minimized using the hybrid particle Swarm optimization with cuckoo search algorithm.

Path1: (N1)-(N1N2)-(N1N2N3N4N5) → 0.5

Path2: (N2)-(N1N2)-(N1N2N3N4N5) → 0.7

Path3: (N3)-(N3N4)-(N3N4N5)-(N1N2N3N4N5) → 0.9

Path4: (N4)-(N3N4)-(N3N4N5)-(N1N2N3N4N5) → 0.2

Path5: (N5)-(N3N4N5)-(N1N2N3N4N5) → 0.4

4.2.1 Minimization Using Hybrid Particle Swarm Optimization with Cuckoo Search

The proposed method employ hybrid particle swarm optimization with cuckoo search for the minimization problem. Particle swarm optimization (PSO) is applied to minimize the path and Cuckoo search is applied to update the initial solution from the particle swarm optimization. Every individual particle i has an arbitrarily initialized position $Yi = (Yi^1, Yi^2, \ldots. Yi^d)$ where Yi^d being its position in the d[th] dimension, velocity, $Vi = (vi^1, vi^2, \ldots. vi^d)$ where vi^d being the velocity in the d[th] dimension,

$$bYyi = (byi^1, byi^2, \dots . byi^d)$$

where byi^d being the best position in the d[th] dimension and $gY = (gy^1, gy^1, \dots . gy^d)$ where gy^d being the global best position in the d[th] dimension in the D-dimensional search space. Some particle can travel in the direction of its personal best position to its best global position in the course of each generation. This process is explained in flowchart Fig.3. The moving process of a swarm particle in the search space is explained as:

$$Vi^d = Vi^d + c_1r_1(bYi^d - Yi^d) + c_2r_2(gY^d - Yi^d) \quad (1)$$

$$Yi^d = Yi^d + \delta Vi^d \quad (2)$$

Where,

c_1, c_2 - constants with the value of 2.0

r_1, r_2-independent random numbers produced in the range[0-1]

Vi^d -Velocity of i[th] particle.

Yi^d-current position of the particle i.

bYi^d-Best fitness value of the particle at the current position.

gY^d-Best fitness value in the swarm.

The flow chart for the hybrid particle swarm optimization with cuckoo search is given below,

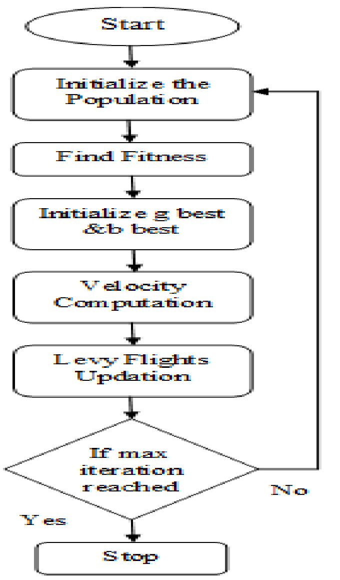

Fig.3 Flowchart for the hybrid particle swarm optimization with cuckoo search

- **Initiate the particle:** For a population size p, build the particles arbitrarily.
- **Discover the fitness function:** The fitness function that should be engaged for the constraints according to the present population. Now eqn. (1) is employed for fitness function calculation.

$$fitness=min \sum_{i=1}^{p} random\ values\ for\ each\ path \quad (3)$$

- **Initialize** gY **and** bY**:** At first the fitness value computed for each of the best one is chosen as the gY and bY value
- **Velocity Computation:** The novel velocity is worked out by means of the below equation. Replace with the c_1 and c_2 values in the velocity equation (1).
- **Swarm Updation:** Work out the fitness function again and revise the gY and bY values. If the novel value is better than the earlier one, substitute the old by the current one. And as well choose the best gY and bY. In our suggested method, cuckoo search algorithm is employed for the swarm Updation.

  Update the initial solution by levy flights. The excellence of the new solution is evaluated and a nest is selected among arbitrarily. If the excellence of new solution in the selected nest is better than the old solutions, it will be replaced by the new solution (Cuckoo). Otherwise, the previous solution is set aside as the best solution. The levy flights employed for cuckoo search algorithm is,

$$Y_i = Y_{i(d+1)} = Y_{i(d)} + \alpha \oplus \text{Levy(n)} \quad (4)$$

- **Criterion to stop:** Carry on till the solution is good enough or maximum iteration is attained finally the proposed method got the optimal path from the hybrid algorithm. The optimal path contains the number of optimal nodes. If one new task is appearing for allocating the resource, the proposed method deploys the minimum path in which the best nodes are selected for the new task. Based on this the proposed method assigns the resource for the available resource with time.

V. CONCLUSION

In this paper, the proposed method discusses resource discovery and resource allocation with hierarchical agglomerative clustering using hybrid artificial bee colony with cuckoo search is implemented in the working platform of JAVA with CloudSim.

VI. REFERENCES

[1] Sheng Di and Cho-Li Wang, Error-Tolerant Resource Allocation and Payment Minimization for Cloud System. IEEE Transactions on Parallel and Distributed Systems 24(6): 1097-1106 (2013).

[2] L. Dhivya and K. Padmaveni, Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment. International Journal of Research in Engineering & Advanced Technology 2(1): 1-4 (2014).

[3] V.Sivaranjani, and R.Jayamala, Optimization of workload prediction based on map Reduce frame work in a cloud system. International Journal of Research in Engineering and Technology 3(3): 264-266 (2014).

[4] Chandrasekhar S. Pawar, Rajnikant B. Wagh, Priority Based Dynamic resource allocation in Cloud Computing, Proceedings of International Symposium on Cloud and Services Computing Pp 1-6 (2012).

[5] Saranya S. and N. Saranya, An Efficient Resource Allocation for Improving Resource Utilization in Self

Organizing Clouds. International Journal of Innovative Research in Computer and Communication Engineering 2(1): 648-1651 (2014).

[6], Pratik P. Pandya, Hitesh A. Bheda, Dynamic Resource Allocation Techniques in Cloud Computing. International Journal of Advance Research in Computer Science and Management Studies 2(1): 559-563 (2014)

[7] Chen He, Ying Lu and David Swanson, Matchmaking: A New MapReduce Scheduling Technique, proceeding of Third IEEE International Conference on Cloud Computing Technology and Science Pp. 40-47 (2011).

[8] Nanthiya D. and P.Keerthika, Load Balancing GridSim Architecture with Fault Tolerance, International conference of information communication and embedded system (ICICAS) Pp. 425-428 (2013).

[9] Haozheng Ren, Yihua Lan and Chao Yin, The Load Balancing Algorithm in Cloud Computing Environment, 2nd International Conference on Computer Science and Network Technology Pp.925-928 (2012).

[10] Hadi Goudarzi and Massoud Pedram, Multi-dimensional SLA-based Resource Allocation for Multitier Cloud Computing Systems, Proceeding of IEEE 4th International Conference on Cloud Computing Pp. 324-331, (2011).

[11] Prasenjit Kumar Patra, Harshpreet Singh and Gurpreet Singh, Fault Tolerance Techniques and Comparative Implementation in Cloud Computing. International Journal of Computer Applications 64(14):37-41(2013).

[12] Chandrashekhar S. Pawar and Rajnikant B. Wagh, Priority Based Dynamic Resource Allocation in Cloud Computing with Modified Waiting Queue, International Conference on Intelligent Systems and Signal Processing (ISSP) Pp. 311-316 (2013)

[13] Dorian Minarolli and Bernd Freisleben, Distributed Resource Allocation to Virtual Machines via Artificial Neural Networks, Proceeding of 22nd Euromicro International Conference on Parallel Distributed and Network-Based Processing Pp. 490-499 (2014)

[14] Wang Xue-yi, Wang Xing-wei and Huang Min, A Multiple Attribute Decision and Bidding based Cloud Resource Dynamic Allocation Method, Proceeding of 8th Annual China Grid Conference Pp. 22-27 (2013).

[15] Goudarzi H., Ghasemazar M. and M.M. Pedram, SLA-based Optimization of Power and Migration Cost in Cloud Computing, Proceeding of 12th ACM International Symposium on Cluster Cloud and Grid Computing (CCGrid) Pp. 172-179 (2012).

[16] Doaa M. and Shawky Performance Evaluation of Dynamic Resource Allocation in Cloud Computing Platforms Using Stochastic Process Algebra, proceeding of 8th International Conference on Computer Engineering & Systems (ICCES) Pp. 39-41 (2013).

[17] Kumar A., E.S.Pilli and R.C.Joshi, An efficient frame work for resource allocation in cloud computing, Proceeding of Fourth International Conference on Computing Communications and Networking Technologies (ICCCNT) Pp. 1-6 (2013).

[18] Zhi Zhou, Fangming Liu, Yong Xu, Ruolan Zou, Hong Xu, John C.S. Lui and Hai Jin, Carbon-aware Load Balancing for Geo-Distributed Cloud Services, Proceeding of IEEE 21st International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems (2013)

[19] Hui Dou, Yong Qi and Peijian Wang, Hybrid Power Control and Electricity Cost Management for Distributed Internet Data Centers in Cloud Computing, Proceeding of 10th Web Information System and Application Conference Pp. 394-399 (2013).

[20] A.M.Aljohani, D.R.W. Holton, I.Awan and J.S.Alanazi, Performance Evaluation of Local and Cloud Deployment of Web Clusters, proceeding of International Conference on Network-Based Information Systems Pp. 244-248,2011.