Asian Journal of Information Technology 15 (22): 4464-4474, 2016
ISSN: 1682-3915

Optimal Resource Discovery and Dynamic Resource Allocation Using Modified Hierarchal Agglomerative Clustering Algorithm and Bi-Objective Hybrid Optimization Algorithm

[1]P. Durgadevi and [2]S. Srinivasan
[1]Department of Information Technology, R.M.K. College of Engineering and Technology, Tamil Nadu, India
[2]Department of Computer Science and Engineering R.M.D Engineering College, Kavarapettai, Tamil Nadu, India

Abstract: The fundamental motive of the resource allocation is to allot the available resource in the most effective manner. It represents the programming of tasks and the resources essential to carryout them, simultaneous taking extreme care with regard to the available resource and the time frame. The vital motive of this investigation is to design a technique for optimal resource discovery and dynamic resource allocation. The innovative technique encompasses two stages such as the resource discovery and resource allocation. For resource discovery the innovative technique utilizes the Modified Hierarchal Agglomerative Clustering Algorithm (MHAC). Based on the MHAC algorithm the suggested tree construction is produced. Thereafter the resources are allocated by the hybrid optimization technique. In the innovative technique, we utilize the Hybrid Artificial Bee Colony and Cuckoo Search algorithm (HABCCS). Here, the artificial bee colony is used to optimize the tree construction path and the cuckoo search is utilized to modify the artificial bee colony algorithm. The optimal path choice is the consequence of the hybrid optimization approach. The new-fangled technique allocates the available resource based on the optimal path.

Key words: Resource allocation, resource discovery, hierarchal agglomerative clustering, artificial bee colony, cuckoo search

INTRODUCTION

Cloud computing is an attractive computing model since it allows for the provision of resources on-demand. Such a process of allocation and reallocation of resources is the key to accommodating unpredictable demands and improving the return on investment from the infrastructure supporting the cloud (Mathapati *et al*., 2012). There is growing interest in improving the energy efficiency of large-scale enterprise data centers and cloud computing environments (Firuzbakht and Bouyer, 2013). Resource discovery is an important process for finding suitable nodes that satisfy application requirements in large loosely coupled distributed systems (Mercy, 2011).

Resource discovery enables applications deployed in heterogeneous large-scale distributed systems to find resources that meet QoS requirements. In particular, most applications need resource requirements to be satisfied simultaneously for multiple resources (Cardosa and Chandra, 2009). Optimal resource allocation and power management in virtualized data centers with time-varying workloads and heterogeneous applications (Firuzbakht and Bouyer, 2013).Clustering aggregation is a kind of formula description about clustering ensample (Zhang *et al*., 2008, 2011, 2012). The goal is to achieve the same quality of resource discovery as a global resource discovery system with full historical node-behavioral knowledge, but to significantly compress the amount of necessary node-behavioral representation data in the system in order to achieve scalability (Kumar and Kaliyamurthie, 2013). To provide efficient clustering without requiring the global knowledge of network by reversing the clustering approach from top-down to bottom up. With the bottom-up approach, sensing nodes build clusters before they select CHs (Jain *et al*., 2014).

The main challenge for any sentence clustering approach is language variability, where the same meaning can be phrased in various ways. For efficient clustering to take place, the inter cluster similarity should be maximum

Corresponding Author: P. Durgadevi, Department of Information Technology, R.M.K College of Engineering and Technology, Tamil Nadu, India

and the intra cluster similarity should be minimum (Priya and Anupriya, 2013). In general, the clustering techniques are broadly divided into partition and hierarchical methods. K-mean is the most popular partition clustering algorithm which is based on centroids (Krishnamoorthy and Kumar, 2012). Agglomerative hierarchical clustering is one of popular clustering techniques (Tamura and Miyamoto, 2014). The attribute based vector space model generalizes standard representations of similarity concept in terms of tree architecture (Liu *et al*., 2010) . The Carrier Aggregation (CA) technology allows scalable expansion of effective bandwidth provided to a user terminal through simultaneous utilization of radio resources across multiple carriers (Azim *et al*., 2010). It takes an input of pair wise data-item similarities and output a hierarchy of the data-items (Dalbouh and Norwawi, 2012). Carrier Aggregation (CA) has been defined as an enabling technology to overcome the spectrum scarcity and fragmentation problem. CA allows a system to aggregate multiple spectrum resources and assign them to a single user in order to provide the sufficient bandwidth for a given service (Zhang *et al*., 2011). K-means imply a K-means algorithm without iteration. An agglomerative hierarchical method with constraints (Obara and Miyamoto, 2012); Bundling, adaptive algorithm and genetic algorithm based clustering aggregation methods. Adaptive algorithm is an algorithm that changes its behavior based on the resources available Once the input profiles have been used to create the graph, the bundling algorithm uses the graph to put the files into bundles method.

Review of recent researches: Skabar and Abdalgader (2013) have proposed an novelty fuzzy clustering algorithm that operates on relational input data, i.e., data in the form of an square matrix of pair wise similarities between data objects. The algorithm uses a graph representation of the data and operates in an expectation maximization framework in which the graph centrality of an object in the graph was interpreted as likelihood. Results of applying the algorithm to sentence clustering tasks demonstrate that the algorithm was capable of identifying overlapping clusters of semantically related sentences and that it was therefore of potential use in an variety of text mining tasks.

Zhang *et al*. (2012) have proposed disadvantages of excessive clustering time because of the possible uneven cluster density. As a result, they present an ASMC algorithm. That proposed algorithm provides adaptability to mobile nodes and has no limitation to the network extensibility. Experiments by the improved J-Sim simulator shown that proposed algorithm evens up the cluster density and extends the network lifetime, compared with the results of LEACH and HEED.

Gao *et al*. (2010) have proposed a hierarchical method; they extended the hierarchical clustering algorithm to cluster fuzzy data for the first time. Finally, that approach has been compared with some of the newly presented methods in the literature. The major advantage of the algorithm was its fault tolerance against noisy samples.

GhasemiGol *et al*. (2010) have proposed an new agglomerative hierarchical clustering algorithm was presented. It was implemented by MapReduce framework. The approach divides the original documents' vectors' set into partitions with the method of initial classification and then distributes the partitions to different data nodes of MapReduce framework. Finally, it processes the documents' vectors in each data node with traditional agglomerative hierarchical clustering algorithm. Benefit from the paralleled procession in each data node, the efficiency of clustering was improved by 86.5%. The speed-up ratio of their framework, which consists of 20 computers was up to 7.414. Compared with traditional k-means and AHC algorithm, the accuracy especially the recall rate of their new approach was improved. Result of experiments shown that their new algorithm implemented on MapReduce framework can apply in large-scale dataset clustering satisfactory.

Lee *et al*. (2014) have proposed an spectrum aggregation or Carrier Aggregation (CA) as referred to in LTE Rel. 10 has required some changes from the baseline LTE Rel. 8 although each CC in LTE-A remains backward compatible with LTE Rel.8. This article provides an review of spectrum aggregation techniques, followed by requirements on Radio Resource Management (RRM) functionality an support of CA. On-going research on the different RRM aspects and algorithms to support CA in LTE Advanced were surveyed.

Wu *et al*. (2012) have proposed a fragment-based approach; a data fragment was any subset of the data that was not split by any of the clustering results. To establish the theoretical bases of the proposed approach, they proved that clustering aggregation can be performed directly on data fragments under two widely used goodness measures for clustering aggregation taken from the literature. Three new clustering aggregation algorithms were described. The experimental results obtained using several public data sets shown that the new algorithms have lower computational complexity than three well-known existing point-based clustering aggregation algorithms (Agglomerative, Furthest and Local Search); nevertheless, the new algorithms do not sacrifice the accuracy.

Yang and Chen (2011) have proposed a temporal data clustering approach via a weighted clustering ensemble on different representations and further propose a useful measure to understand clustering ensemble algorithms based on a formal clustering ensemble analysis. Simulations show that their approach yields favorite results for a variety of temporal data clustering tasks in terms of clustering quality and model selection. As a generic framework, their weighted clustering ensemble approach allows other validation criteria to be incorporated directly to generate a new weighting scheme as long as they better reflect the intrinsic structure underlying a data set.

Problem definition: In this study briefly discussed the problem definition shown below:

- While using resource usage histogram provides a means to capture representation of individual nodes
- The dynamic resource usage pattern and enables the satisfaction of statistical resource requirements potentially creates a scalability problem
- The major problem is time complexity
- Agglomerative hierarchical clustering is not good at handling huge data sets because of the computational complexity
- Page Rank must be applied to each cluster in each EM cycle and this can lead to long convergence times if the problem involves a large number of objects and/or clusters
- Outlier handling it may happen that a node doesn't belong to any cluster and it resides in a cluster of its own. This makes the clustering inefficient
- In clustering, the exact meaning of each cluster may not be obvious at real run time

MATERIALS AND METHODS

Location of the resources and their allocation in a cost-conscious manner is the vital motive of the innovative technique which flows through two different phases like the resource discovery and the resource allocation. In the preliminary phase, the representative resource usage allocation for a group of nodes with associated resource usage patterns is performed as resource bundle which is used to locate a group of nodes satisfying a general obligation. Now, the clustering-based resource aggregation like the Modified Hierarchal Agglomerative Clustering Algorithm (MHAC) is launched to achieve the compact depiction of a set of similarly behaving nodes for scalability. In the subsequent phase for dynamic resource allocation, we use the bi-objective hybrid optimization technique dependent upon the artificial bee colony coupled with the cuckoo search algorithm.

Resource discovery phase: At the outset, initialize the specification of each node in the resource. The innovative technique generates the tree structure in accordance with the node specification. Here node contains some specification like memory, size, IP address, location id and storage. At present, the adapted hierarchical agglomerative clustering technique is used for the tree structure which is concisely detailed.

Modified hierarchical agglomerative clustering algorithm: The hierarchical clustering successively segregates a dataset with a specific distance measure. In this sequential separation procedure, an algorithm builds nested partitions layer by layer by means of grouping objects into a tree of clusters entirely in accordance with the distance measure without the necessity to recognize the number of clusters well-ahead. There are two techniques to a tree of clusters such as the Agglomerative Hierarchical clustering algorithm or AGNES (bottom up) and the Divisive Hierarchical clustering algorithm or DIANA (top down). Both these techniques are exactly the reverse of each other. The top down approach assumes that all objects in a dataset are initially in a distinct cluster and subsequently the cluster is repeatedly segmented into smaller and smaller clusters until a stopping criterion is satisfied. In contrast, the bottom up approach, well-known as the agglomerative technique, assumes that all objects in a dataset are atomic clusters of a particular element. Subsequently, all the atomic clusters unite to form into a bigger cluster. Our novel approach follows the agglomerative hierarchical clustering technique. With an eye on augmenting the efficiency in performance of the conventional agglomerative technique, the adapted agglomerative hierarchical clustering algorithm is employed. At this juncture, clustering of each node is in accordance with the minimum distance with the IP address and location id. The specific procedure of the adapted hierarchical agglomerative clustering algorithm with

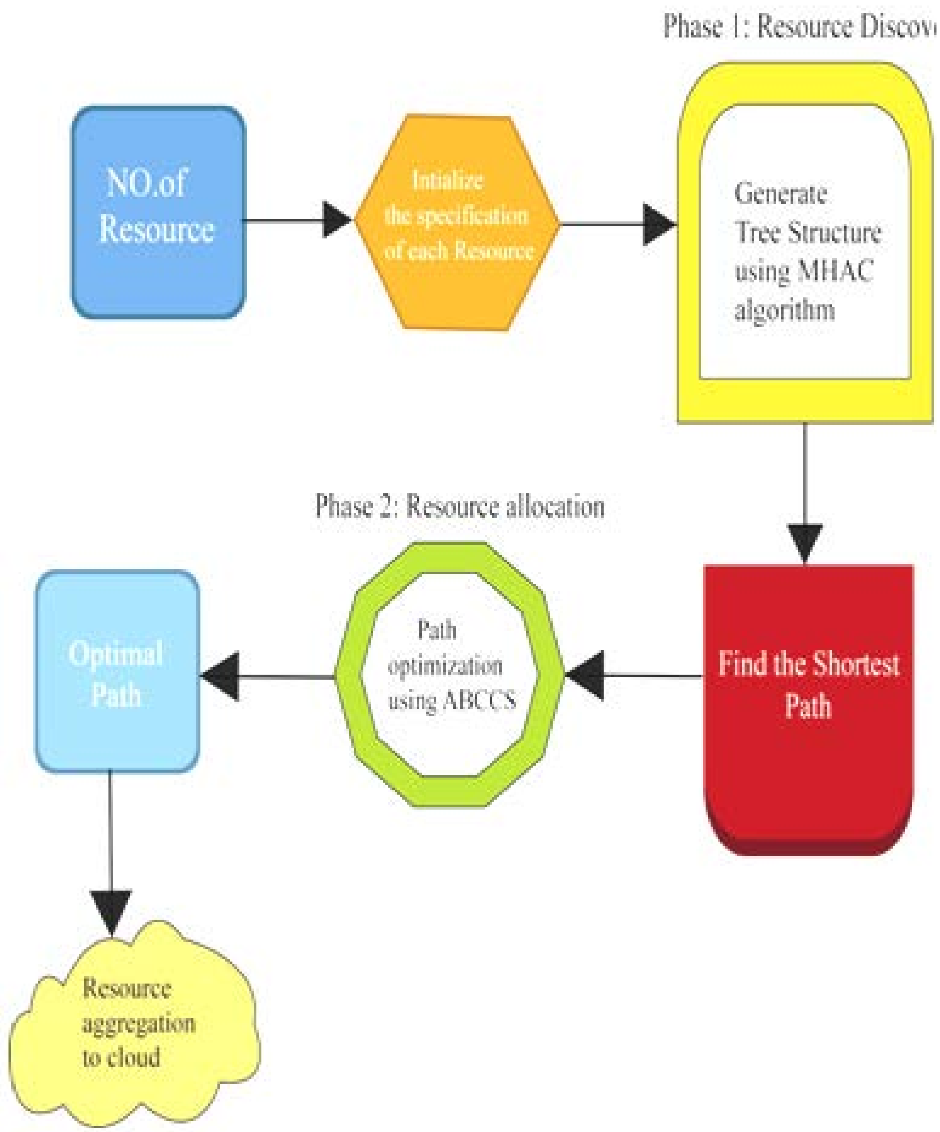

Fig. 1: Block diagram of the proposed method

example is shown as follows algorithm.

The overall procedure of hierarchical agglomerative clustering algorithm:

Input: No of nodes with IP address and Location id for each node
Output: Cluster hierachy or dendromgram
Intilization: Number of node (N_i, where I = 1,2,..k)
IP address and Location id
disjoint clustering level L(0) = 0
sequence number n = 0.

Start
Step 1: Assess all pair-wise distance among nodes
Add each nodesas its own cluster
Step 2: Erect distance matrix by means of the distance values
Dc [(N_1), (N_2)]
Step 3: pair the node with ther shortest distance
N = [$N_1 \cup N_2$]//Marge the node
Step 4: incraese n = n+1,
Set the level ti L(n) = dc [(N_1), (N_2)]
Step 5: Revise the distance matrix
Distance between the new clusters
Dc [(N_3), (N)] = min (Dc [(N_3), (N_1)], Dc [(N_3), (N_2)])
Step 6: Replicate till the distance matrix is decreased to a single element
Stop

Example: Let us consider a resource with five nodes. Now each node is characterized as N1-N5. At the outset, we use IP address and location id for each five nodes. Thereafter, we proceed to create the tree according to the IP address and location id of each node with the help of agglomerative hierarchical clustering technique (Fig. 2).

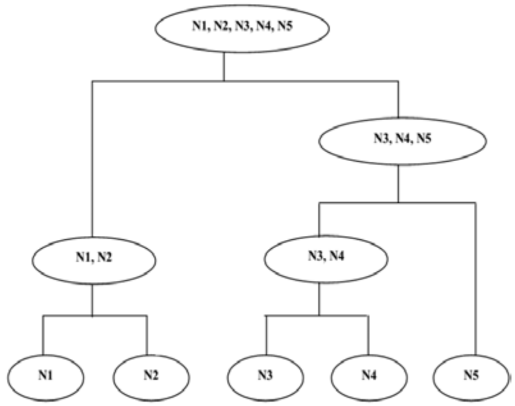

Fig. 2: Example of modified agglomerative hierarchical clustering algorithm

Now the IP address and location id of the node is selected. The clustering among the nodes is in accordance with the distance between the IP address and location id. Here, nodes N3 and N4 represent the minimum distance so that they are clustered together (N3, N4). Likewise, as nodes N1 and N2 characterize the minimum distance they are grouped together (N1, N2). Subsequent, the new group (N3, N4) and node N5 consist of the novel group or cluster (N3-N5) is created. In the long run, all the nodes are grouped together to form a single cluster (N1-N5). Now, it is our turn to move to the phase two after the tree creation.

Resource allocation phase: In this phase, in accordance with the hybrid optimization technique, the resource is assigned. To allocate the resource for the task, we find the shortest path in the tree construction with an eye on locating g the best path. After finding the number of shortest path in the tree construction; then those number of shortest path is optimized with the help of the hybrid optimization technique. Let us, consider a tree construction with a number of shortest paths which is exhibited in Fig. 3. The tree comprises n number of leaf nodes and depending on their minimum distance the leaf nodes are grouped together. To arrive at the parent node each leaf node moves in their path. The path linked to each leaf node and their tree structure is illustrated below:

Path1: (N1)-(N1N2)-(N1N2N3N4N5)

Path2: (N2)-(N1N2)-(N1N2N3N4N5)

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

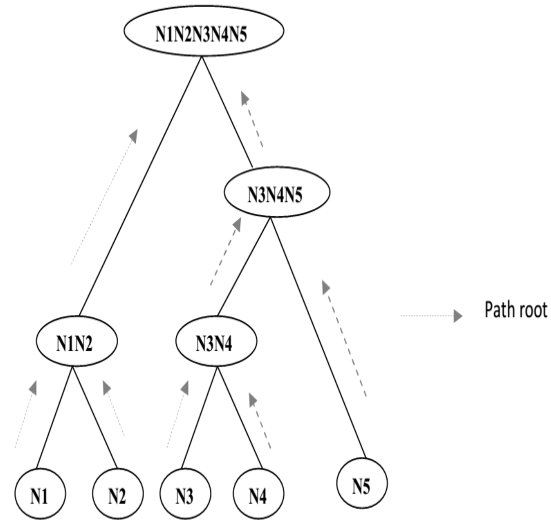

Fig. 3: The tree construction with path root

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

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

In the ground-breaking technique, we find the shortest path for the entire leaf node. Subsequently all the shortest paths are optimized with the help of the hybrid artificial bee colony coupled with cuckoo search algorithm. The detailed procedure of the hybrid optimization is explained as. The various paths are achieved by employing the above two phases. For allotting the resources, an optimal path is required as the time utilized is high while employing several paths. From the number of paths, the optimal path is found out by means of a hybrid optimization algorithm. The hybrid optimization represents the blend of the artificial Bee Colony Algorithm (ABC) and the Cuckoo Search (CS) algorithm which offers the optimal paths for resource allocation in accordance with the best nodes.

Optimization using hybrid artificial bee colony algorithm with Cuckoo search: The innovative technique utilizes the hybrid artificial bee colony algorithm with cuckoo search for the optimization issue. Here, the artificial Bee Colony Algorithm (ABC) is used to optimize the path and the Cuckoo search is utilized to revise the initial solution from the artificial bee colony algorithm. The ABC technique constitutes a swarm based meta-heuristic algorithm. The number of shortest paths is furnished input artificial bee colony algorithm. Each food source is exhibited as the path. The ABC technique comprises three

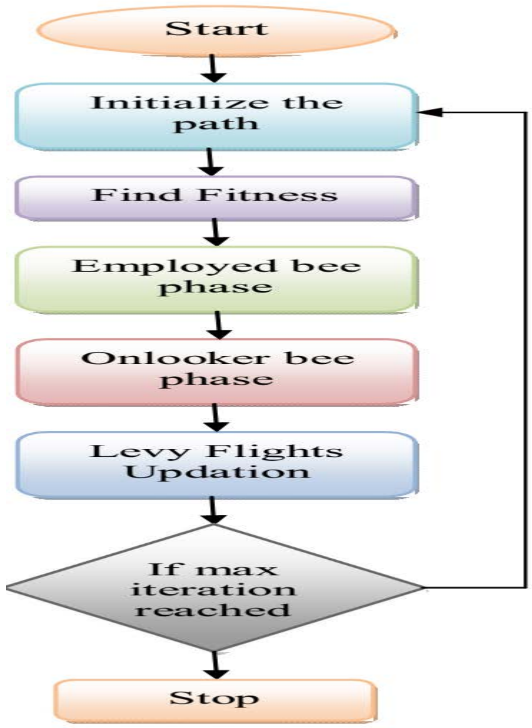

Fig. 4 : Flowchart for the hybrid optimization algorithm

modules such as the employed bees, onlooker bees and the scout bees. The employed bees are united with the path and hand over the nectar quality of the path to onlooker bee. The nectar quality of each path is shown as the fitness value. The onlooker bees watch the dance of the employed bees. The employed bees whose path is given up emerge as the Scout and go in search of new path randomly. The entire process is illustrated in flowchart vide Fig. 4. The modus operandi of the innovative hybrid optimization technique is illustrated as:

Initialize the shortest path: Each individual bee follows a randomly initialized shortest path X_i where $i = 1, 2,\ldots,d$ represents shortest path in the Nth dimension.

Find the fitness function: The fitness function selection has to be utilized for the restraints based on the current path. Equation 1 is used for estimating the fitness function:

$$\text{fitness} = \begin{cases} \min \text{distance} \\ \min \text{cost} \\ \max \text{size} \end{cases} \tag{1}$$

Employed bee phase: In the employed bee phase, the new paths are created by means of the equation shown:

$$NF_{i,j} = X_{i,j} + \gamma_{ij}\left(X_{i,j} - X_{k,j}\right) \tag{2}$$

Where:
k and j = Represents the random selected index
γ = Denotes the random number [-1, 1]
$Nf_{i,j}$ = Characterizes the new food source or new path

Thereafter, the fitness is evaluated for every new path. Out of them, that best path is chosen. In other words, the path which has the greatest fitness value by means of the greedy selection process is chosen as the best path and thereafter the probability of the chosen path is estimated with the help of Eq. 3:

$$\text{probability} = \frac{\text{fitness}_i}{\sum_{i=1}^{n} \text{fitness}_i} \tag{3}$$

Onlooker bee phase: When the probability of the chosen path is found out, onlooker bee is evaluated for the purpose of creating the new path for the onlooker bees from the initial path based on the probability value. Thereafter, the fitness for the new path is estimated and the greedy selection is initiated so as to choose the best path.

Scout bee phase: If any abandoned path is existing; it is substituted with the new path located by scouts by means of Eq. 3 and the fitness value is estimated. If the new value is superior to the previous value, then the latter is substituted by the former value. In the innovative technique, the cuckoo search algorithm is utilized for the scout bee function.

Now, the initial solution is updated by levy flights. The quality of the new path is estimated and a nest is chosen randomly. If the quality of new path in the chosen nest is superior to the old path, it is substituted by the new path (Cuckoo). Or else, the old path continues to remain as the best path. The levy flights used for the cuckoo search technique are given by Eq. 4:

$$X_i^{*} = X_i^{new} = X_i + \alpha \oplus \text{Levy}(n) \tag{4}$$

Criterion to stop: The process is continued till the path is excellent or the maximum iteration is achieved. In the long run, we arrive at the optimal shortest path from the hybrid artificial bee colony algorithm with cuckoo search. Now each optimal path is home to a number of optimal nodes. If one new task arrives for the allocation of the resource, the innovative technique employs the optimal path in which the best nodes are chosen for the new task. With a view to choose the best nodes, the executed techniques deploy the resource cost, time and memory capacity. Depending on this, the innovative technique allots the resource for the available resource. Finally the optimal node is aggregate in to the cloud using resource converging algorithm. It is detail illustrated in the below study algorithm.. It is detail illustrated in the algorithm.

The overall procedure of resource converging alogrithm; Alogorthim for creating private pool:

Input: Any private cloud (G_n)
Output: private resource pool $_{Private}$ (G_n)
Intilization: resource set (rx(x) = null), node queue (Q = null) node acces vector (lag = 0)
Step 1: take one random node is starting node (n_r) then put n_r int Q and set flag (n_r) = 1
Step 2: if Q = null then assigne node as $P_{private}$ (G_n)
Step 3: Q = n_{rs}; Get the entire node with axcces vector flag = 0 from the neighborhood and put them into Q then modified the node acces vector flag = 1.
Step 4: If the resource edge set = null, then go to step 2
Step 5: Then search the resource edge set and collect the corresponding resource edge into resource set according to their resource type.
Step 6: If the searching the completed then go the step. 2
Step 7: $P_{private}$ (G_n) = {resource set | type ∈ type id}

Alogorthim for Converting private cloud into golbal
Input: private cloud set (G = G_1, G_2...G_n)
Output: Golbal categortized resource pool $P_{PGolbal}$ (G_n)
Intilization: Golbal categortized resource set (Grs(x) = null), node queue (Q_1 = null) node, private Cloud acces vector (Flag = 0)
Step 1: take one random private cloud is starting cloud (G_r) then put G_r into Q and set flag (G_r) = 1
Step 2: if Q_1 = null then assigne private cloud as P_{golbal} (G_n)
Step 3: Q_1 = G_{rs}; Get the entire cloud with axcces vector flag = 0 from the neighborhood and put them into Q_1 then modified the cloud acces vector flag = 1.
Step 4: in accordance with the respective typr, merge the private cloud into golbal.
Step 5: If the resource merging pf private cloud is completed then go to step 2, else go to step 4
Step 6: P_{golbal} (G_n) = {golabl cloud | type ∈ type id}

RESULTS AND DISCUSSION

The innovative resource discovery and resource allocation with modified hierarchical agglomerative clustering employing hybrid artificial bee colony with cuckoo search is performed in the working platform of JAVA with CloudSim. The cost and memory values are also estimated and its average value is contrasted with that of the current method. The table appearing below illustrates the cost value of our Hybrid Artificial Bee Colony algorithm and Cuckoo Search algorithm (HABCCS).

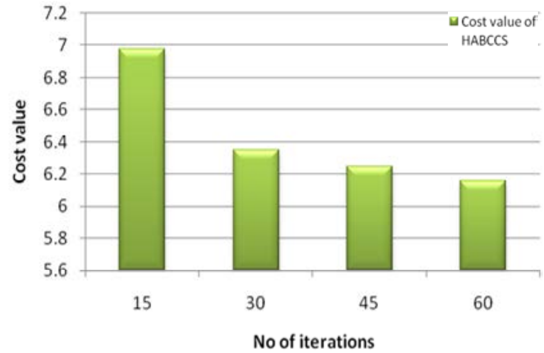

Fig. 5: Cost value of proposed HABCCS

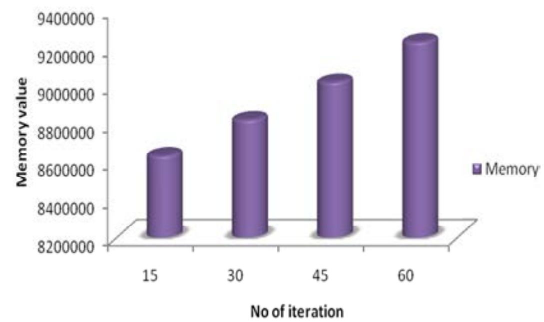

Fig. 6: Memory value of proposed HABCCS

Now the number of iteration of the innovative HABCCS technique is altered and the cost value of for each iteration is calculated. Table 1 illustrates that the hybrid artificial bee colony and the cuckoo search have the least cost value when it completes the iteration 60. The average cost value of the innovative HABCCS technique is 6.43, thus, it is able to attain the least cost value. The pictorial depiction of the cost value for the innovative technique is furnished Fig. 5.

Table 2 represents the memory value of iteration in the innovative technique. In the 15th iteration the memory value of the hybrid artificial bee colony and cuckoo search is 8632145 bit and in the 30th iteration the memory allotted for the HABCCS is 8821419 bit. So the innovative technique utilizes the least memory allocation for the number of iteration for each resource.

The pictorial depiction of Table 2 is given. Here x axis represents the number of iterations and y axis represents the memory value (Fig. 6).

Table 3 reveals the time taken for each iteration based on seconds. To finish the 15th iteration the innovative technique spends 8003 sec. The corresponding value for finishing the 45th iteration is 18624 sec. The novel approach finishes the 60th iteration in 21646 sec. The average time duration taken to finish the

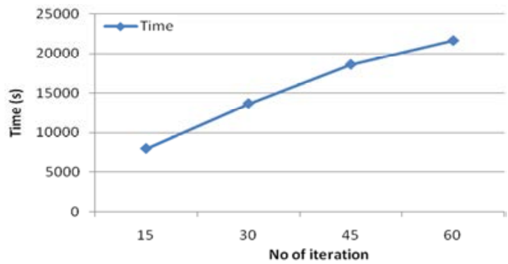

Fig. 7: Time taken for the proposed method

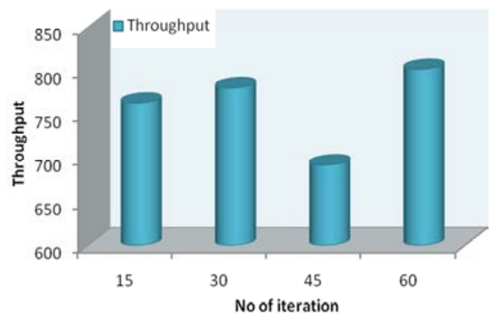

Fig. 8: Throughput value of proposed HABCCS

Table 1: No of iteration with the cost value of the HABCCS

No. of iterations	Cost value of HABCCS
15	6.97951
30	6.34975
45	6.24774
60	6.15475

Table 2: The memory value of the HABCCS

No. of iterations	Memory
15	8632145
30	8821419
45	9019461
60	9231417

Table 3: Time taken for the proposed HABCCS

No. of iterations	Time
15	8003
30	13684
45	18624
60	21646

HABCCS technique, the time taken is about 15489.25 sec. The graphical illustration is exhibited in Fig. 7.

The general throughput value of the HABCCS is 759.2325. The tabulation of the innovative throughput value and their related graphical illustration is presented in Fig. 8 and Table 5. The throughput value ranges from 0-763.15 in the 15th iteration. When the HABCCS technique arrives at the 45th iteration the throughput value ranges from 0-692.14. Hence, by modifying the number of iteration the throughput value of the HABCCS also undergoes change.

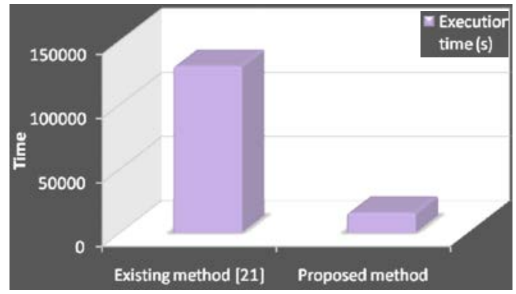

Fig. 9: Comparative analysis of proposed method

Table 4: Throughput value of HABCCS

No. of iterations	Throughput
15	763.15
30	780.34
45	692.14
60	801.40

Table 5: Comparison result of proposed method

Metrics	Existing method (Wu *et al*., 2012)	Proposed method
Execution time(s)	129855	15489.25

In the number of iteration the cost value for the innovative technique is achieves minimum value and the time utilization of each iteration is shows the minimum time.

Comparative analysis: Here the existing works are compared with our proposed work, in order to prove the proposed work is better one. For this existing (Wu *et al*., 2012) is taken to compare the result with our method. The following table is shown the comparative result. The graphical representation of comparative analysis is shown in Fig. 9. Here, Fig. 9 illustrates the graphical representation of comparative analysis. It is shown in Fig. 9 and Table 5.

From our results of comparison, we can say that our proposed research rescues the execution time. The existing research (Wu *et al*., 2012) is taken 129855 seconds to complete the clustering process but our proposed HABCCS takes minimum execution time. It takes nearly 15489.25 sec to complete the clustering process (Table 4 and 5).

From these existing works, we can say that our proposed reduces the execution time when compared to the existing method. The screen shot for our proposed HABCCS algorithm is shown in Fig. 10. At first the output of our technique is look like Fig. 10. It is demonstrate in beneath, Then we have to upload the file for allocating the resource. Figure 11 represents the file selection process of HABCCS method.

After that, the file has been selected for uploading. Based on these, we have allocated the resource node for the file by using HABCCS algorithm. In the resource node, the data are stored and illustrated in the following Fig. 12.

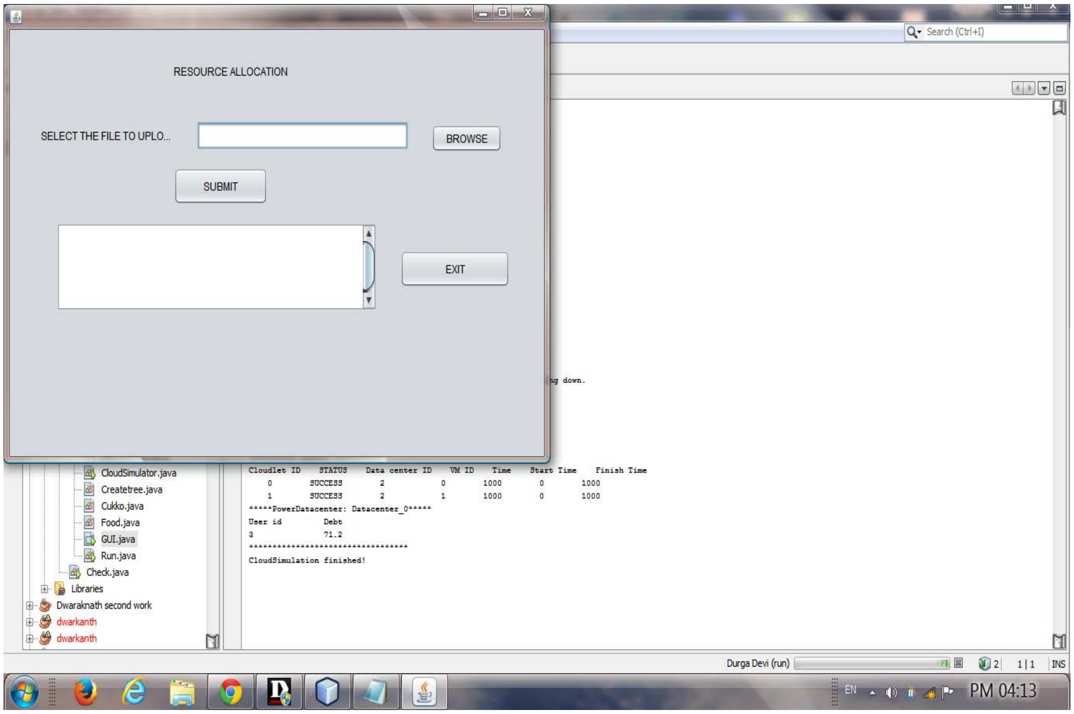

Fig. 10: The output screen shot

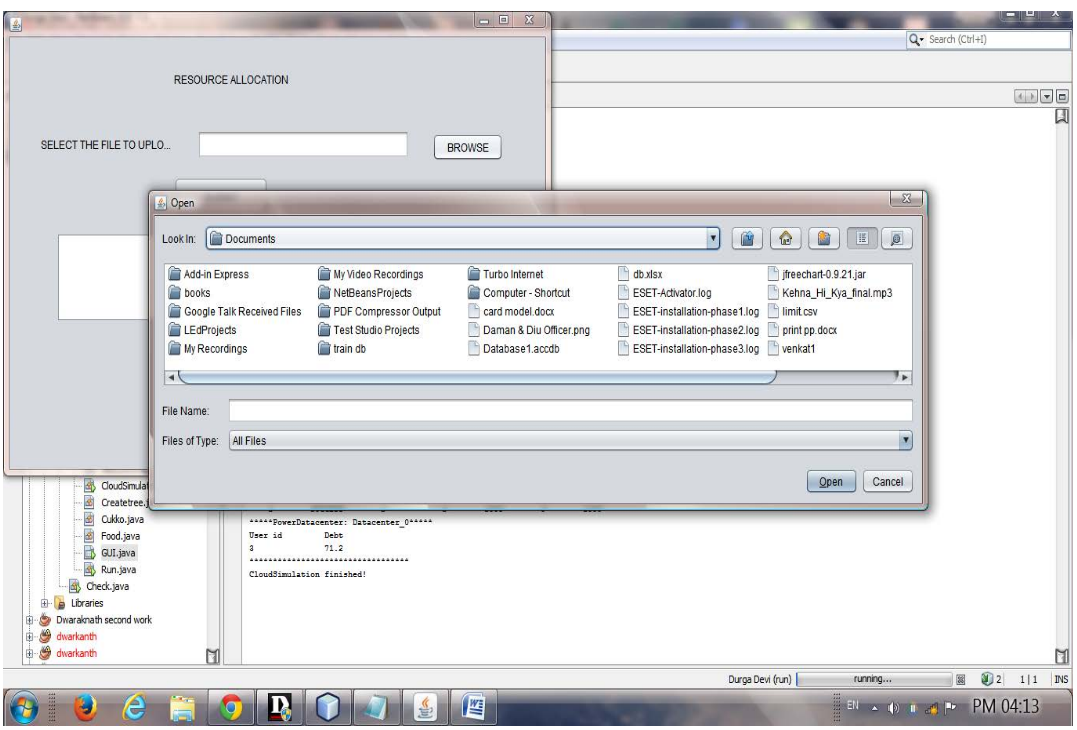

Fig. 11: File selection dialogbox

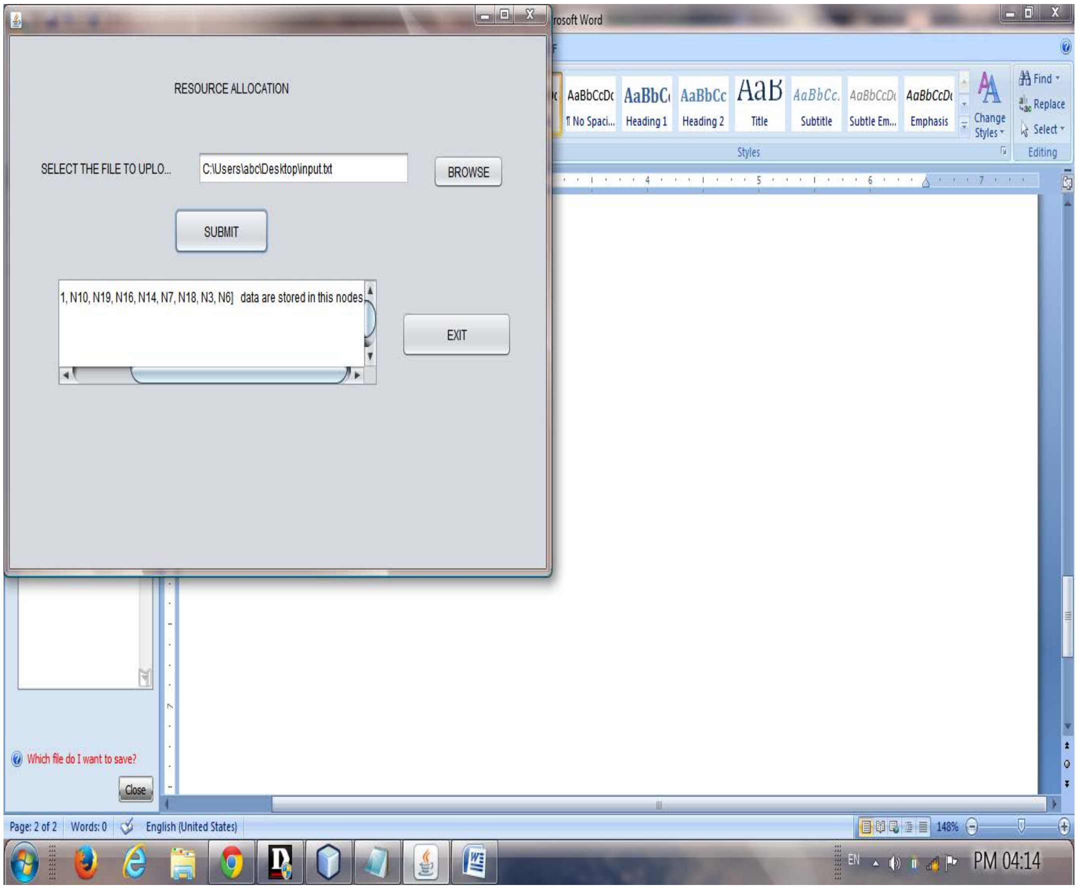

Fig. 12: The output window of proposed HABCCS

CONCLUSION

In this document, the optimal resource discovery and dynamic resource allocation are proposed. At the outset, the resources are located in accordance with the adapted Hierarchical Agglomerative Clustering Algorithm (MHAC) and the tree construction is carried out. Subsequently, the resources are allocated with the help of the hybrid optimization algorithm and in this investigation; the artificial bee colony and cuckoo search are deemed to be hybrid. In our epoch-making technique, the Artificial Bee Colony (ABC) is used to optimize the path and the Cuckoo Search (CS) is utilized to modify the population of the artificial bee colony. It is clear from the captivating outcomes that the available resources are allocated in the most effective manner with least computation duration. It is hoped that the upcoming investigator will have ample opportunities to perform with their own optimization approach and scale newer heights of excellence in performance.

REFERENCES

Azim, M.A., S. Moad and N. Bouabdallah, 2010. SAG: Smart aggregation technique for continuous-monitoring in wireless sensor networks. Proceedings of the 2010 IEEE International Conference on Communications (ICC), May 23-27, 2010, IEEE, New York, USA., ISBN: 978-1-4244-6402-9, pp: 1-6.

Cardosa, M. and A. Chandra, 2009. HiDRA: Statistical multi-dimensional resource discovery for large-scale systems. Proceedings of the 17th International Workshop on Quality of Service (IWQoS), July 13-15, 2009, IEEE, New York, USA., ISBN: 978-1-4244-3875-4, pp: 1-9.

Dalbouh, H.A. and N.M. Norwawi, 2012. Improvement on agglomerative hierarchical clustering algorithm based on tree data structure with bidirectional approach. Proceedings of the 2012 3rd International Conference on Intelligent Systems Modelling and Simulation, February 8-10, 2012, IEEE, New York, USA., ISBN: 978-1-4673-0886-1, pp: 25-30.

Firuzbakht, A. and A. Bouyer, 2013. An optimal algorithm based on gridding and clustering for decrease energy consumption in WSN. Proceedings of the 36th International Conference on Convention on Information and Communication Technology Electronics and Microelectronics (MIPRO), May 20-24, 2013, MIPRO Company, Chiayi, Taiwan, ISBN: 978-953-233-073-1, pp: 474-478.

Gao, H., J. Jiang, L. She and Y. Fu, 2010. A new agglomerative hierarchical clustering algorithm implementation based on the map reduce framework. Int. J. Digital Content Technol. Appl. (JDCTA.), 4: 95-100.

GhasemiGol, M., H.S. Yazdi and R. Monsefi, 2010. A new hierarchical clustering algorithm on fuzzy data (fhca). Int. J. Comput. Electr. Eng., 2: 134-140.

Jain, T.K., D.S. Saini and S.V. Bhooshan, 2014. Performance analysis of hierarchical agglomerative clustering in a wireless sensor network using quantitative data. Proceedings of the 2014 International Conference on Information Systems and Computer Networks (ISCON), March 1-2, 2014, IEEE, New York, USA., ISBN: 978-1-4799-2981-8, pp: 99-104.

Krishnamoorthy, R. and S.S. Kumar, 2012. New optimized agglomerative clustering algorithm using multilevel threshold for finding optimum number of clusters on large data set. Proceedings of the 2012 International Conference on Emerging Trends in Science Engineering and Technology (INCOSET), December 13-14, 2012, IEEE, New York, USA., ISBN: 978-1-4673-5141-6, pp: 121-129.

Kumar, M.B. and D.K. Kaliyamurthie, 2013. A novel resource distributed discovery and management in grid computing. Int. J. Comput. Sci. Mob. Comput., 2: 623-627.

Lee, H., S. Vahid and K. Moessner, 2014. A survey of radio resource management for spectrum aggregation in LTE-advanced. IEEE Commun. Sur. Tutorials, 16: 745-760.

Liu, H., H. Bao, J. Wang and D. Xu, 2010. A novel vector space model for tree based concept similarity measurement. Proceedings of the 2010 2nd IEEE International Conference on Information Management and Engineering (ICIME), April 16-18, 2010, IEEE, New York, USA., ISBN: 978-1-4244-5263-7, pp: 144-148.

Mathapati, B.S., S.R. Patil and V.D. Mytri, 2012. Energy efficient reliable data aggregation technique for wireless sensor networks. Proceedings of the 2012 International Conference on Computing Sciences (ICCS), September 14-15, 2012, IEEE, New York, USA., ISBN: 978-1-4673-2647-6, pp: 153-158.

Mercy, M.H., 2011. Resource overbooking: Using aggregation profiling in large scale resource discovery. Int. J. Engi. Trends Technol., 1: 52-54.

Obara, N. and S. Miyamoto, 2012. A method of two-stage clustering with constraints using agglomerative hierarchical algorithm and one-pass k-means. Proceedings of the 13th International Symposium and 2012 Joint 6th International Conference on Soft Computing and Intelligent Systems (SCIS) and Advanced Intelligent Systems (ISIS), November 20-24, 2012, IEEE, New York, USA., ISBN: 978-1-4673-2742-8, pp: 1540-1544.

Priya, G.K. and G. Anupriya, 2013. Clustering sentence level-text using fuzzy hierarchical algorithm. Proceedings of the 2013 International Conference on Human Computer Interactions (ICHCI.), August 23-24, 2013, IEEE, New York, USA., ISBN: 978-1-4673-5703-6, pp: 1-8.

Skabar, A. and K. Abdalgader, 2013. Clustering sentence-level text using a novel fuzzy relational clustering algorithm. IEEE Trans. Knowledge Data Eng., 25: 62-75.

Tamura, Y. and S. Miyamoto, 2014. A method of two stage clustering using agglomerative hierarchical algorithms with one-pass k-means++ or k-median++. Proceedings of the IEEE International Conference on Granular Computing (GrC), October 20-24, 2014, IEEE, New York, USA., ISBN: 978-1-4799-5464-3, pp: 281-285.

Wu, O., W. Hu, S.J. Maybank, M. Zhu and B. Li, 2012. Efficient clustering aggregation based on data fragments. IEEE. Trans. Syst. Man Cybern., 42: 913-926.

Yang, Y. and K. Chen, 2011. Temporal data clustering via weighted clustering ensemble with different representations. IEEE Trans. Knowledge Data Eng., 23: 307-320.

Zhang, L., K. Zheng, W. Wang and L. Huang, 2011. Performance analysis on carrier scheduling schemes in the long-term evolution-advanced system with carrier aggregation. IET Commun., 5: 612-619.

Zhang, Y., K. Li, H. Gu and D. Yang, 2012. Adaptive split-and-merge clustering algorithm for wireless sensor networks. Proc. Eng., 29: 3547-3551.

Zhang, Z., H. Cheng, S. Zhang, W. Chen and Q. Fang, 2008. Clustering aggregation based on genetic algorithm for documents clustering. Proceedings of the 2008 IEEE Congress on Evolutionary Computation and IEEE World Congress on Computational Intelligence, June 1-6, 2008, IEEE, New York, USA., ISBN: 978-1-4244-1822-0, pp: 3156-3161.