Journal of Engineering and Applied Sciences 12 (Special Issue 2): 6325-6331, 2017
ISSN: 1816-949X

Heuristic Method for Cloud Resource Consolidation with ECRC Algorithm

G. Soniya Priyatharsini and N. Malarvizhi
Department of Computer Science and Engineering,
Veltech Dr. RR and Dr. SR Technical University, Avadi, Chennai, Tamil Nadu, India

Abstract: Now a days the very inspired and the famous technology in computing is the cloud computing technology. One of the main area is the server consolidation in cloud computing. This is an area in which researchers rarely touched. Residual resource defragmentation is the method of decreasing fragmentation of resources which is residuals. The remaining resources will be less useful or useless. This is an obstacle in resource fragmentation. This leads to the datacenter provider by high expenditure. In each consolidation level the residual resource fragmentation reduced. Thus the dynamic resource provisioning reduced its migrations. In this research, it proposes a resource provisioning dynamically and allocation technique. The proposed method concludes by 3 phases. Firstly, by using optimization algorithm, physical servers which are active are selected. This includes, binary Cuckoo search algorithm to select the optimal server. The fitness to select the optimal server is memory and cost. Next step is that the calculation of the optimal server's maximum utilization of resources. The resources are categorized based on the type. Next is the allocation of these resources to the appropriate physical servers or virtual servers. Thus the data defragmentation in data centers is reduced. Finally, ECRC (Enhanced Cloud Resource Consolidating) assists in the scheduling and allocation of the identified resources. The implementation research for this will be done using java in cloudsim.

Key words: Server sprawl, resource utilization, server consolidation, resource defragmentation, implementation, optimal server

INTRODUCTION

Cloud computing can be explained as the services and applications that accessed by common internet protocols and researches using virtualized network resources and networking standards on a distributed network. The word "CLOUD" is the very popular buzz in the recent years. For internet, it is historically identified as a metaphor. Cloud computing is everywhere. In 2008, according to Larry (CEO of Oracle) stated that even women's fashion will be less fashion driven than the computer industry. Globalization of computing assets may be the biggest contribution the cloud has made to date. Cloud computing allows us to access applications device most often, this will be distant datacenter. From this, it explains that these cloud providers handle the cost of servers. They also manage the software updates. It is very much comfortable for the remotely working people who are travelling can just simply enter in and start using their application in any other places.

Consolidation of servers: The above described consolidation of the servers is an accession of the cloud server's resources to an effective usage. The total numbers of the located servers are reduced. Here, the consolidation of server process responded to the identified server sprawl problem. Server sprawl problem is that the servers which are rarely used occupies more spaces and utilize many resources is called server sprawl. Figure 1 dictates the consolidated process of servers.

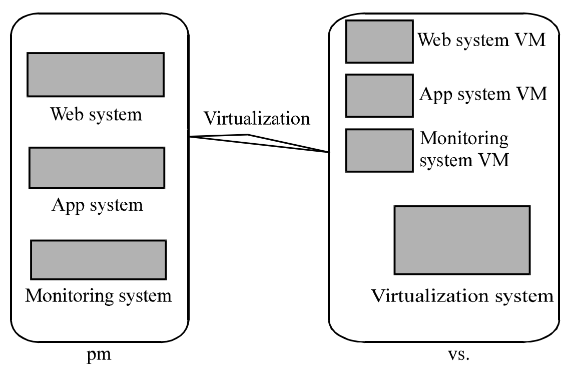

Fig. 1: Process of server consolidation

Resource pooling of residuals: An IT term resource pooling is used in the cloud computing environment which describe a provider serve multiple clients with

Corresponding Author: G. Soniya Priyatharsini, Department of Computer Science and Engineering,
Veltech Dr. RR and Dr. SR Technical University, Avadi, Chennai, Tamil Nadu, India

Table 1: Resource allocation inputs

Parameters	Service provider	Data owner
Application requirements	No	Yes
Resource status	Yes	No
Available resources	Yes	No
Provider offerings	Yes	No

scalable and provisional services. The client need can be adjusted according to their needs, without doing any changes. In the cloud environment when it needs to allocates the resources, the cloud provider activities are integrated with the resource allocations strategy. Resource allocation for optimal strategies as to abstain the below listed terms.

Over provisioning: This condition reaches whenever the consumer gets surplus resource than the demanded one.

Resource dissent: Situation appears when more than one application tries to use one same resource.

Resource scarcity: This appears when the resources are limited.

Resources fragmenting: This condition occurs whenever the resource is isolated.

Lesser provisioning: This attains if the demand amount of resources is higher than the allocated resources. To overcome the allocation of the resource strategy, the below listed tables are needed (Table 1).

Auto scaling: Auto scaling is the process of the adjusting the size of the instances which was handled for the client's load. Auto scaling groups are created by the cloud data owner with the collections of instances. In each auto scaling group the minimum number of instances can be specified by the user. The size of the group should be go below. This can be assured by the auto scaling. The maximum number of instances can be specified by the user in each auto scaling group. It also ensures the size should not go up. Auto scaling ensures that the user's group has this many instances.

Literature review: In some of the earlier study, the researchers projected the server consolidation process in their research. The goal of the Khanna's (Vyas and Chauhan, 2016) algorithm is to reach the sever consolidation. The resources like memory and CPU are considered and the number of migrations are identified in the entropy, algorithm. Memory is considered as a resource and the dirty page bit is considered as the metrics in the Miyako Dori's (Himthani, 2016) algorithm. Memory is considered as the resource in the memory buddies (Rao and Thilagam, 2015). The goal of the Sandpiper (Thakur *et al*., 2013) algorithm is hotspot. It also identifies the memory, CPU and the network as the resources. The Sercon algorithm (Hermenier *et al*., 2009) identifies CPU and memory as its metrics. Table 2 contains the details of the above paragraph in short.

Server sprawl is the problem identified. Many of the techniques such as improvised first fit decreasing (Akiyama *et al*., 2012), harmonic constrained approach, first fit and best fit decreasing (Wood *et al*., 2009a, b), Sercon (Khanna *et al*., 2006) and best fit decreasing (Ho *et al*., 2011). These are the applications which are used in maximum case of the bin packing algorithms. Some of the techniques like genetic algorithm based method (Bein *et al*., 2011), p-mapper (Ajiro and Tanaka, 2007), 2-phase optimization method (Beloglazov and Buyya, 2010a, b), dynamic round robin approach (Marzolla *et al*., 2011), control theoretic solution (Lin *et al*., 2011), adaptive threshold based methods (Beloglazov and Buyya, 2010a, b) provide various techniques and methods for controlling the consumption of the energy and reducing methods for power like formulation of LP.

Problem statement: From the studies related to this study, it is decided that the fragmentation of the residual resources are not much concentrated. Reducing the physical servers can be concentrated by some of the papers. Lower power consumption will be the result for this. The resources are reused when the residual resources are concentrated. Cost will be reduced with this. It gives much cost reduction. When the situation occurs in cloud hub in which the resources which are resources are available at sufficient state is called the residual resource fragmentation. The fragmented resources were rendering them unusable across distributed areas. Thus the reducing process of the resource fragmentation is known to be resource defragmentation. This process improves the utilization of the resources only. Memory, CPU and network bandwidth utilization weren't improved by this process. Simultaneously a single scalar is used. If the destination PM is decided with the help of the given scalar score, it directs the residual resources to be fragmented. The unused resources were not used for the next procedures. Thus, the resource fragmentation problem leads to the residual resources less useful or useless. Thus, the datacenter provider adds cost to the reduced resource utilization. So, here in this study it proposes a technique which reduces the residual resource fragmentation. It allocates and schedules the resources for processes.

Table 2: Server consolidation; existing works

Algorithms used	Metrics concentrated	Resource considered	Goal of the algorithm
Khanna's algorithm (Vyas and Chauhan, 2016)	Residual capacity, variance	CPU, memory	Server consolidation
Entropy	No. of migrations	CPU, memory	Server consolidation
Miyako Dori (Himthani, 2016)	Dirty page bit	Memory	Server consolidation
Memory buddies (Rao and Thilagam, 2014)	Memory size	Memory	Server consolidation and Hotspot mitigation
Sandpiper (Thakur *et al*., 2013)	Volume	Memory CPU and network	Hotspot mitigation
Sercon (Murtazaev and Oh, 2011)	Memory, CPU and load capacity	Memory and CPU	Server consolidation

MATERIALS AND METHODS

Solution methodology: For the above problem mentioned there is a solution in this study. That is explained in three steps; in the first stage by using optimization algorithm the active servers are identified, i.e. by utilizing Cuckoo search algorithm the optimal servers can be selected. To identify the optimal servers the cost will be the fitness. Binary Cuckoo search algorithm is used to identify the physical machines which are active. The binary Cuckoo search algorithm can be explained as the binary version of the Cuckoo search algorithm. Nozarian explained this algorithm where Cuckoo search algorithm is developed and explained in 2009 by the Xin-She Yang. Three rules are executed to the simplicity in describing Cuckoo search; every Cuckoo bird lay its egg in the same instant also it put its egg where it randomly chosen nest. The best eggs will be carried out to the next generation over with highest quality of eggs by the best nests. Once the total counting of the found nests are fixed, then the host bird discover the Cuckoo bird's egg and the probability of this is given as pa∈[0,1].

While considering rn(i), rm(i) and rc(i) as the network bandwidth requirement, memory requirement and cost for the virtual machines vm v of i, here, i can be represented from 1-n. Consider Rn, Rnm and Rc as the threshold of the consumption of the network bandwidth, memory and the CPU correspondingly in the physical machine PMs. Consider N(j) be represented as the current state of the PMs, the group of virtual machines available in the physical machines of the jth position. In which j = 1, 2, 3, till m. The value of N(j) threshold is found by using the consolidation process of the virtual machine sets and the physical machines, i.e., pr(ji) where ji gives the values from 1-l. This gives the consolidation of the server's work. After processing this consolidation of the server process, the following Eq. 1-3 are solved:

$$\sum_{K\in N(j)} rc(k) \le RC \tag{1}$$

$$\sum_{K\in N(j)} rc(k) \le Rm \tag{2}$$

Table 3: Conditions for PM migrations

Types	Utilization of relative resources
1	baw>=memo>=CPU
2	mem>=baw>=CPU
3	CPU>=memo>=bw
4	baw>=CPU >=memo
5	CPU >=baw>=memo
6	memo>=CPU >=baw

$$\sum_{k\in N(j)} rn(k) \le Rn \tag{3}$$

Where the value of j extends from 1-l. Secondly, from optimized servers the utmost utilized resources are calculated. On the basis of the type the identified resources are categorized and then allocated to the physical servers which are suitable. It also may be the virtual machines. It reduces the fragmentation of the data from the cloud hub.

Then Virtual Machines (VMs) and Physical Machines (PMs) are migrated with the following conditions.

The exchange of virtual machines between the physical machines should improvise resource fragmenting process. The physical machine from where the group of virtual machines are chosed ought to satisfy the 3 consumption process R_{mc}. The circumstances are given in Table 3.

Here, the memory, network bandwidth and the Central Processing Unit are doing the main roles. Here, all reasonable situations for the resource usage are covered by all above six conditions. The process of defragmentation can be found using the Eq. 4 which is given as:

$$\text{Defragment} = \frac{\sum_{i=1}^{m}\left(T_j - C_{ij}\right)^2}{\sum_{i=1}^{m}\left(T_j - C_{ij}\right)} \tag{4}$$

Where:

m = Explains the active P's number

T_{ji} = Explains jth resource's threshold

C_{ij} = Explains the jth resource in the ith P's current utilization

If we need less resource fragmentation then the defragment value should be high. Thus, residual resource defragmentation value matters for the defragment, i.e.:

$$\text{Total defragmtT} = \text{defragmt CPU} + \text{defragmt MEM} + \text{defragmt BW} \quad (5)$$

Thirdly, the scheduling and allocation of the identified resources are done using Enhanced Cloud Resource Consolidating (ECRC).

In second phase optimized servers are calculated with the maximum utilization. Using ECRC (Enhanced Cloud Resource Consolidating) technique the resources are allocated and scheduled. This research extended the research of cloud resource consolidation algorithm. From the study 20, the easiest path is found by using the ant colony algorithm. Then the random walk method was explained in this algorithm of the Levy's flight (Verma *et al*., 2008). This type of an algorithm for efficient routing is useful in the network. The size of the data's defragmentation is the fitness for the selection process of the resources. Maximum extent usage of the left over resources is because the residual resource defragmentation is carried out by the server consolidation process. So, according to the client's requisite resources are selected. If the available resources are high while compared to the user's requirement, then the score value is fixed. On its basis the fitness is calculated. The data is defragmented based on this defragmented value. An algorithm workflow of an ECRC algorithm is explained as.

Algoritham 1:

```
Output; best node from the given better nodes
public class ECRC
{public static void update()
{arraylists<arraylists <doubles>> tem = new arraylists <arraylists <
doubles >> ()
for ( int ij = 0; i < veloc.sizes (); i++)
tem.adds (veloc.gets (ij))
veloc.clear()
for (int ij = 0; i < temp.sizes (); ij ++)
{arraylist < double> t = new arraylists<double s> ()
for ( int jk = 0; j < temp.gets (ij).sizes (); jk ++)
{double d = tem.gets (ij). get (jk)+(0.2*(genran()*((initial.get
(pb).get(j))-(initial.get(i).get(j)))))+(0.2* (genran()*(gbarray.get(j)-
(initial.get(i).get(j)))))
double d1 = tem.get(i). get(j)+(0.2*(genran() *((initial. get(pbh). get(j))-
(initial.get(i).get(j)))))+(0.2*(genran()*(gbharray.get(j)-(initial.get(i). get(j)))))
if(d<d1)
t.add(d)
else
t.add(d1);} velo.add(t);}
for (int ij = 0; i< veloc.sizes ( ); ij ++ )
system.outs.println ( veloc.gets ( ij )+" nv ")
} }
```

Figure 2 explains the workflow of the ECRC algorithm. In ECRC technique, the fitness is the size of the data defragmentation. On the basis of the client's task requirement the resources are selected. Maximum extent of the leftover resources is used, if the residual resource defragmentation is carried out by the server consolidation process. Thus the user's requirement on the resources is selected. Defragmentation can be done on the basis of the optimized worth. Algorithm which describes the proposed ECRC can be explained in the following.

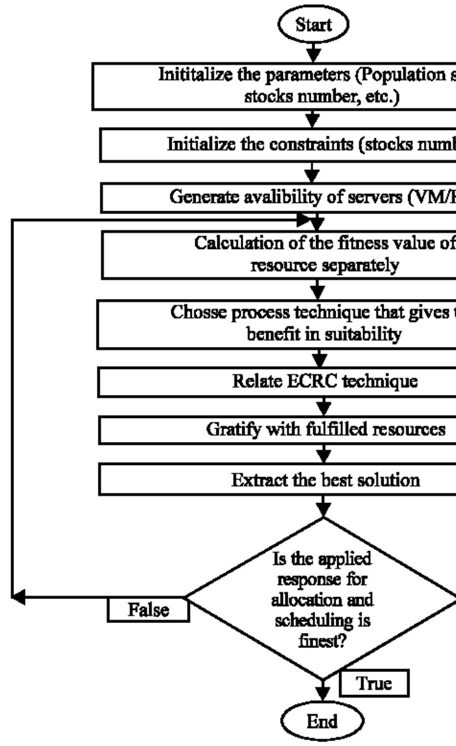

Fig. 2: ECRC technique's working procedure

Workflow of ECRC algorithm

Proposed algorithm 2:

```
Output; best node from the given better nodes
public class ECRC
{public static void update()
{arraylists<arraylists<doubles>>tem = new arraylists<arraylists<doubles>>
( )
for (int ij = 0; ij< veloc.sizes ( ); ij++ )
temp. adds ( veloc. gets (ij) )
velo.clear()
for ( int ij = 0; ij< temp.sizes ( ); ij++)
{ arraylist < double> t = new arraylists < doubles > ( )
for( int jk = 0; j<temps.gets ( ij ).sizes ( ); jk++ )
{double d = temps.gets ( ij ). gets ( jk ) +(0.2*(genran()*((initial.get
(pb).get(j))-(initial.get(i).get(j)))))+(0.2* (genran()*(gbarray.get(j)-
(initial.get(i).get(j)))))
double d1 = tem. get(i). get(j)+(0.2 *(genran()* ((initial.get(pbh).get(j))-
(initial.get(i).get(j)))))+(0.2*(genran()*(gbharray.get(j)-(initial.get(i). get(j)))))
if(d<d1)
t.add(d)
else
```

```
t.add(d1);} veloc.add(t);}
for (int ij = 0; i < veloc.sizes ( ); ij + +)
system. outs. println ( veloc. get ( ij )+" nv");}}
```

In the direction of finding the better nodes process the ECRC technique is used. Whenever the better nodes are identified, the next process is to find the best node from the better nodes. It can be done with respect to the client's requirement. The price is used to find the thickness if two or more nodes found out. Server sprawl problem can be facing many changes with the help of this technique. The resources can be provisioned by the ecrc algorithm which can be provides the resources as per the clien's needs in the various stages of provisioning. The implementation of this proposed research will be done using java and in cloudsim.

RESULTS AND DICUSSION

Experimental study 1 (energy consumption): Figure 3 illustrates the cost reduction. Data center energy consumption dataset is explained in Table 4. Optimized cloud resource provisioning, optimal virtual machine placements also the different workloads considered by the various algorithms are considered. Figure 3 in the above illustrates that the values for the ecrc algorithm is not much varies from the existing algorithms. They are almost same. So, it can be concluded that this technique is useful in the energy consumption process. So, the identified ecrc technique successfully reduced the energy consumption in the cloud data center. It also reduces the cost comparatively with the OCRP along with OVMP.

Experimental study 2 (defragmentation of data): Figure 4 explains the data fragmentation of the identified physical servers. It compares with the already existing algorithms like OVMP and OCRP for server consolidation. Table 5 explains the values for the data fragmentation. In each consolidation interval, the defragmentations of the residual resources are comparatively high for ECRC technique when compared to the existing techniques like OCRP and OVMP. So, from the experimental study it can be concluded that the proposed ECRC technique provides better result while comparing with the existing algorithms. The retrieved result can be gained only because of the defragment value. Whenever the value of the defragment is higher, then the defragment is lesser.

Experimental study 3(data allocation and scheduling): Here, in this research (Fig. 5 and 6) to reach the convergence state, the virtual machines are allocated with respect to the different algorithms. Initially 10 virtual

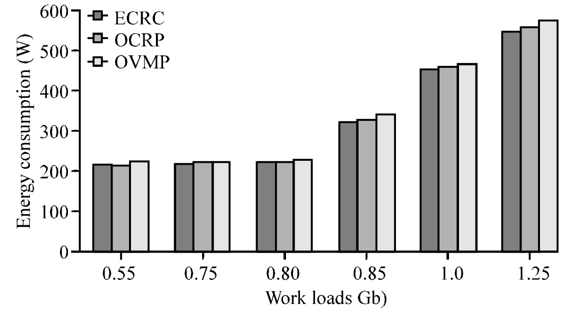

Fig. 3: Consumption of energy using various algorithms

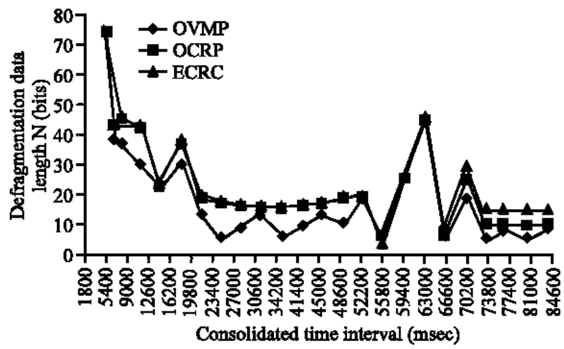

Fig. 4: At different consolidation intervals residual resource defragmentation

Table 4: Energy consumption datasets

Workload in size	ECRC	OCRP	OVMP	ECRC on OCRP (%)	ECRC on OVMP (%)
0.55 (GB)	187.2	185.4	192.3	0.009615	0.02724
0.75 (GB)	189.7	192.3	195.0	0.013710	0.02794
0.80 (GB)	192.1	194.3	197.4	0.011450	0.02759
0.85 (GB)	278.2	284.7	292.5	0.023360	0.05140
1 (GB)	389.4	395.2	409.4	0.014890	0.05136
1.25 (GB)	467.9	478.8	491.9	0.023300	0.05129

Table 5: Defragmentation datasets

Consolidated interval	OCRP	OVMP	ECRC
1800	64	71	65
5400	31	28	32
9000	30	24	30
12600	21	20	22
16200	26	25	28
19800	17	15	18
23400	15	10	16
27000	11	12	15
30600	13	14	15
34200	14	10	15
41400	16	12	16
45000	17	14	17
48600	17	13	18
52200	18	18	18
55800	11	10	11
59400	20	20	21
63000	30	30	32
66600	15	12	19
70200	21	18	23
73800	14	10	15
77400	14	11	15
81000	14	10	15
84600	14	11	14

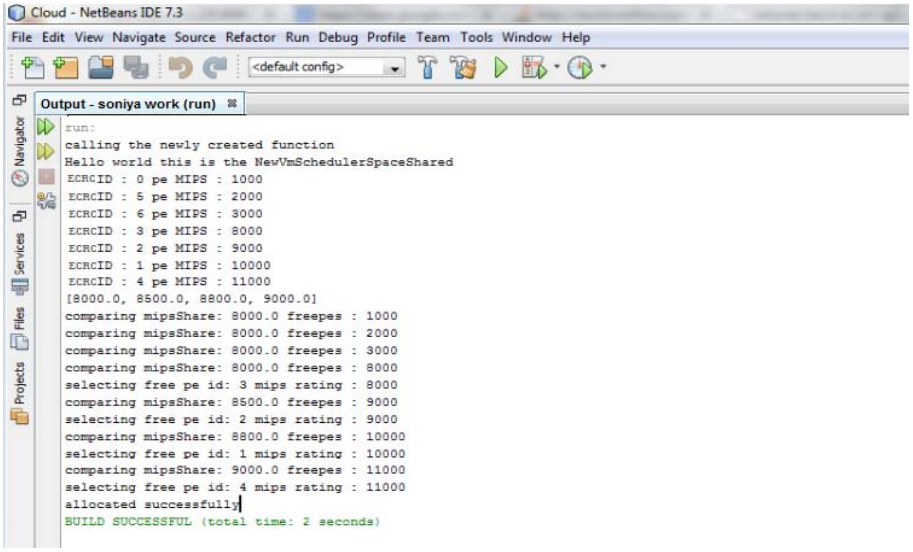

Fig. 5: Allocated machines as per the request

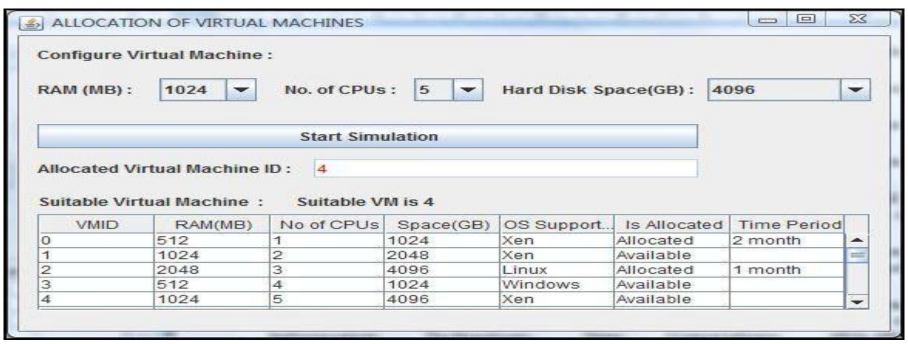

Fig. 6: List of allocated machines for scheduling

machines are allocated using the OVMP technique and the ECRC technique. Here, it gives 3 outputs for each algorithm. It proves that it will be change for different techniques. Likewise here the experiment is done for various allocations of virtual machines. That is from 10-100 virtual machines. This allocation explains the convergence state for the virtual machines.

Once the machines are allocated as per the request of the data user, the machines are get ready for the scheduled process. This scheduling happens with the basis of the memory space available in the allocated machines and the cost. Thus the fitness is calculated by using the cost and the memory. The nearest memory space and the cost can be considered for the procedures.

CONCLUSION

With the help of the proposed ECRC technique, the server can be optimized for the allocation according to the user's need. Thus the related server is identified as per the user's request. The best fit for the task is recognized from the listed and identified top nodes. Here, thickness of the values is assigned by the defragmentation. The results may differ from the already presented techniques using ecrc for residual resource allocation. It is concluded as follows from the observations done. By reducing the energy consumption the ECRC reduced the cost expenditure. By using these techniques each consolidation interval the fragmentation varies. The Enhanced cloud resource consolidation algorithm schedules more virtual machine migrations, the proposed approach considers the unused space in resources in an efficient manner.

IMPLEMENTATIONS

Further, implementations could be done in scheduling migrations to consider the data of resource utilization. Thus, the proposed research is efficient in cost and memory space.

REFERENCES

Ajiro, Y. and A. Tanaka, 2007. Improving packing algorithms for server consolidation. Proceedings of the 33rd International Conference on Computer Measurement Group, December 2-7, 2007, CMG, San Diego, California, USA., pp: 399-406.

Akiyama, S., T. Hirofuchi, R. Takano and S. Honiden, 2012. Miyakodori: A memory reusing mechanism for dynamic VM consolidation. Proceedings of the 2012 IEEE 5th International Conference on Cloud Computing (CLOUD), June 24-29, 2012, IEEE, Honolulu, Hawaii, ISBN:978-1-4673-2892-0, pp: 606-613.

Bein, D., W.W. Bein and S. Venigella, 2011. Cloud S torage and Online Bin Packing. In: Intelligent Distributed Computing-IDC, Brazier, F.M.T., N. Kees, P. Gregor, W. Martijn and B. Costin (Eds.). Springer, Berlin, Germany, ISBN:978-3-642-24012-6, pp: 63-68.

Beloglazov, A. and R. Buyya, 2010a. Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers. Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and E-Science, November 29-December 3, 2010, ACM, Bangalore, India, ISBN:978-1-4503-0453-5, pp: 41-46.

Beloglazov, A. and R. Buyya, 2010b. Energy efficient allocation of virtual machines in cloud data centers. Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), May 17-20, 2010, IEEE, Melbourne, Victoria, Australia, ISBN:978-1-4244-6987-1, pp: 577-578.

Hermenier, F., X. Lorca, J.M. Menaud, G. Muller and J. Lawall, 2009. Entropy: A consolidation manager for clusters. Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE '09), March 11-13, 2009, ACM, Washington, DC., USA., ISBN:978-1-60558-375-4, pp: 41-50.

Himthani, M.P., 2016. Efficient technique for allocation of processing elements to virtual machines in cloud environment. Intl. J. Comput. Sci. Netw. Secur., 16: 107-111.

Ho, Y., P. Liu and J.J. Wu, 2011. Server consolidation algorithms with bounded migration cost and performance guarantees in cloud computing. Proceedings of the 2011 4th IEEE International Conference on Utility and Cloud Computing (UCC), December 5-8, 2011, IEEE, Taipei, Taiwan, ISBN:978-1-4577-2116-8, pp: 154-161.

Khanna, G., K. Beaty, G. Kar and A. Kochut, 2006. Application performance management in virtualized server environments. Proceedings of the 10th IEEE/IFIP Symposium on Network Operations and Management (NOMS 2006), April 3-7, 2006, IEEE, Vancouver, British Columbia, Canada, ISBN:1-4244-0142-9, pp: 373-381.

Lin, C.C., P. Liu and J.J. Wu, 2011. Energy-aware virtual machine dynamic provision and scheduling for cloud computing. Proceedings of the 2011 IEEE International Conference on Cloud Computing (CLOUD), July 4-9, 2011, IEEE, Washington, DC., USA., ISBN:978-1-4577-0836-7, pp: 736-737.

Marzolla, M., O. Babaoglu and F. Panzieri, 2011. Server consolidation in clouds through gossiping. Proceedings of the 2011 IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM), June 20-24, 2011, IEEE, Lucca, Italy, ISBN:978-1-4577-0352-2, pp: 1-6.

Murtazaev, A. and S. Oh, 2011. Sercon: Server consolidation algorithm using live migration of virtual machines for green computing. IETE. Tech. Rev., 28: 212-231.

Rao, K.S. and P.S. Thilagam, 2015. Heuristics based server consolidation with residual resource defragmentation in cloud data centers. Future Gener. Comput. Syst., 50: 87-98.

Thakur, S., A. Kalia and J. Thakur, 2013. Server consolidation algorithms for cloud computing environment: A review. Intl. J. Adv. Res. Comput. Sci. Software Eng., 3: 379-384.

Verma, A., P. Ahuja and A. Neogi, 2008. PMapper: Power and Migration Cost Aware Application Placement in Virtualized Systems. In: Middleware 2008, Valerie, I. and R. Schantz (Eds.). Springer, Berlin, Germany, ISBN: 978-3-540-89855-9, pp: 243-264.

Vyas, N. and A. Chauhan, 2016. A survey on virtual machine migration techniques in cloud computing. Intl. J. Appl. Innovation Eng. Manage., 5: 297-300.

Wood, T., G. Tarasuk-Levin, P. Shenoy, P. Desnoyers, E. Cecchet and M.D. Corner, 2009a. Memory buddies: Exploiting page sharing for smart colocation in virtualized data centers. Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, March 11-13, 2009, ACM, Washington, DC., USA., ISBN: 978-1-60558-375-4, pp: 31-40.

Wood, T., P. Shenoy, A. Venkataramani and the M. Yousif, 2009b. Sandpiper: Black-box and gray-box resource management for the virtual machines. Comput. The Networks, 53: 2923-2938.