Analysis of Enhanced Genetic learning-based particle swarm optimization in Cloud Computing Environment

Harpal Singh[1] Pawan Kumar[2]
[1]*Research Scholar, NIILM University, Kaithal*
[2]*Assoc. Professor, NIILM University, Kaithal*

Abstract - Cloud computing has revolutionized the ease with which millions of details can be managed in the digital and internet era. It acts as an intermediary between applications and data centers as a model for distributing services over the Internet. Orders from many customers are managed and fulfilled by the cloudIt is typically a "pay per base" network because facilities are allocated to subscribers on a case-by-case basis. Customers can use the cloud to access a variety of data resources that are tailored to their needs.

I. VIRTUAL MACHINE CONSOLIDATION IN CLOUD ENVIRONMENT

A server may be configured to support many virtual machines (VMs) with the assistance of virtualization. Virtualization allows for the scaling up and down of the resource capabilities of a virtual machine (VM) as well as the consolidation of servers through the selective movement of VMs across hosts. Through power management, this consolidation technique decreases the number of hosts by either turning them off or transitioning them to a low-power sleep state that is subsequently left un- used, lowering data center power usage overall. Only a percentage of the computer machine's potential power is utilized, and many systems operate with a low average system load in most cases. Many of the physical resources and electrical power have been depleted. As a result, instead of using real computers or hosts that are only half utilized, numerous virtual machines (VMs) may be crammed onto a small number of strong hosts by balancing them. During server consolidation, all virtual machines (VMs) running on various under-utilized servers are transferred using a live VM migration technique. All the servers that are not in use will be put into a power-savingmode. Live VM migration guarantees that service downtime is kept to a bare minimum, as is the time required for the transfer. In recent years, great emphasis has been placed on building data centers in a energy-efficient and reliable way. This technique can take you to several different places. To increase power economy, it is standard practice to slow down the speed of the CPU and switch off portions of the hardware components. Server consolidation and power-aware VM mapping are techniques for reducing power usage by putting unneeded machines into a power- saving mode while they are not in use. In the proposed research effort, dynamic consolidation of servers is done in a cloud environment to reduce power consumption, resource waste, and the number of virtual machine migrations in data centers while maintaining high performance.

II. ENHANCED GENETIC LEARNING PARTICLE SWARM OPTIMIZATION

The authors have an idea of optimizing the genetic algorithm with the help of particle swarm optimization [21]. The authors have suggested that the genetic algorithm will initially help the particle swarm optimization to find the near-optimal solution. Once the near-optimal solution of the genetic algorithm is collected, these results make the particles of the PSO algorithm. Both algorithms continue to work in parallel.

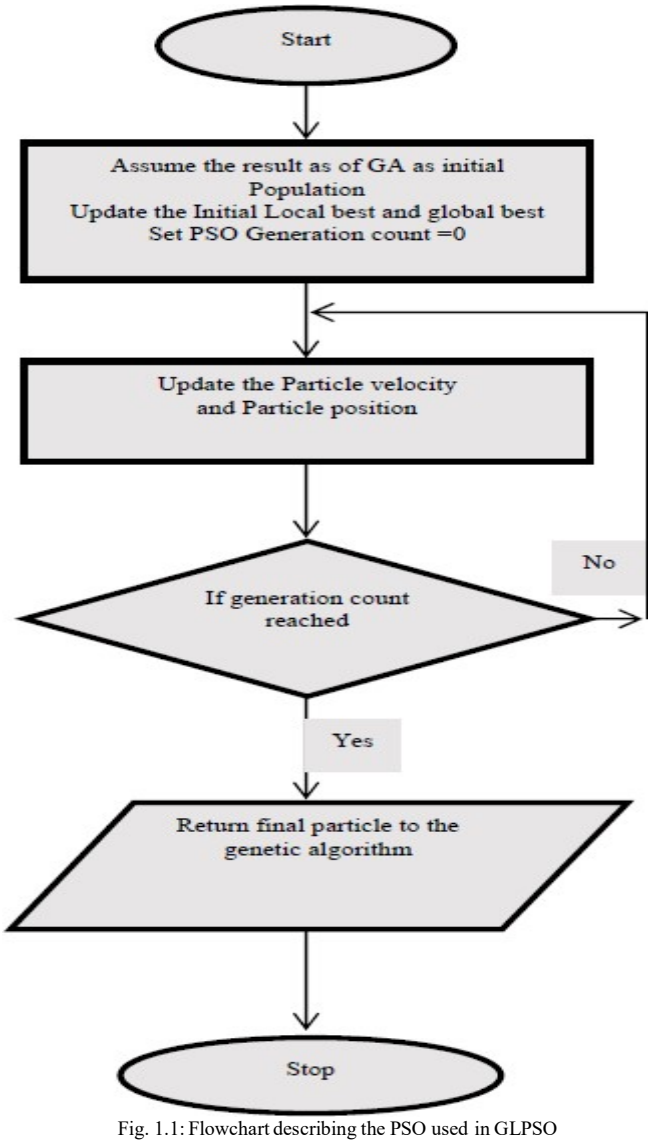

Fig. 1.1: Flowchart describing the PSO used in GLPSO

The traditional solutions obtained from PSO are precise to the nearby optimal solution, as shown in fig. 1.1. The genetic algorithm, on the other side, generates the new solution by performing a heuristic search and maintains the diversity in the solutions.

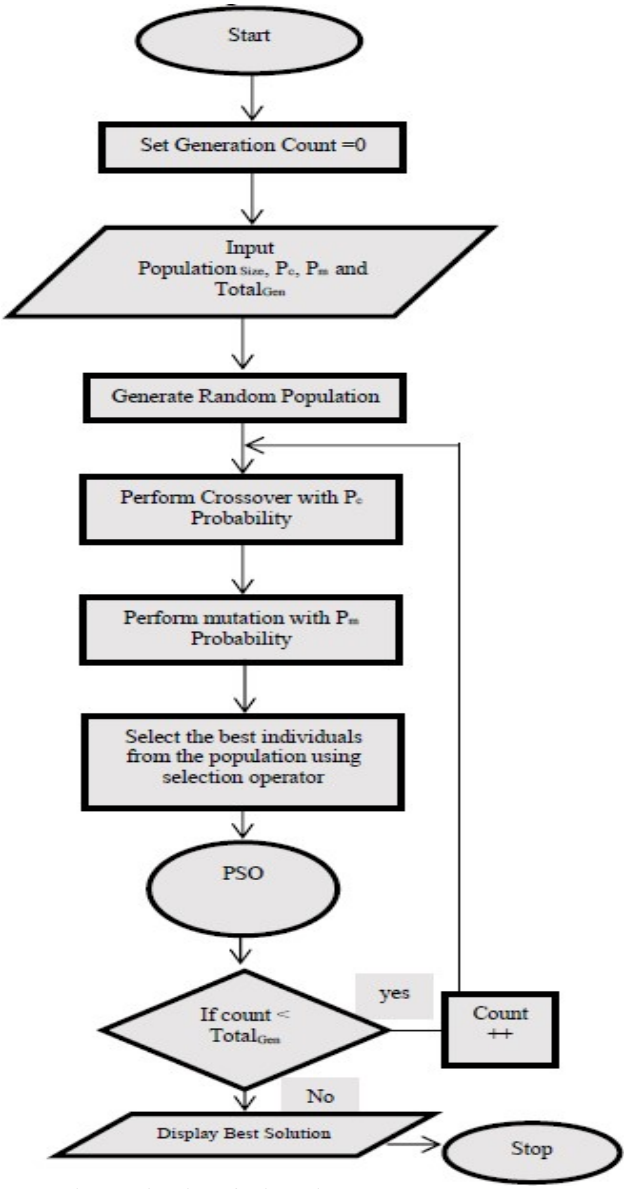

Fig. 1.2: Flowchart of Enhanced GLPSO

The GL-PSO algorithm has two variants proposed by the authors. In the first variant, PSO and GA are working in parallel. In the second version, PSO the part of the genetic algorithm. In the proposed algorithm, The PSO working on the results obtained from the genetic algorithm with a different number of iterations; Figure 3a details the flowchart of the PSO algorithm. The major drawback of this approach is that the PSO algorithm s not getting its complete power to explore the solutions as in every iteration new population is served, which deteriorates the overall performance of the particle swarm optimization. In our approach, the modified things that instead of using PSO as a single operator, it provides the Genetic algorithm results are behaving as the input to the PSO for *m* iterations, as shown in fig. The computational cost will increase, but the solution obtained will be more diverse and precise. The detailed algorithm is as follows.

Algorithm 1: Enhanced GLPSO Algorithm

Algorithm 2: (Best Solution) := **EGLPSO** (N, $Total_{Gen}$, P_c, P_m, $Total_{GenPSO}$)	
1.	$Gen_{count} \leftarrow 0$
2.	(Pop_{Gen}):=Generaterandompopulation(N); //Generate the initial population
3.	**while** $Gen_{Count} < Total_{Gen}$
4.	g_{best}:=Assignglobalbest();//Assign Global best
5.	($Temp_{gen}$):=Crossover(P_c.Pop_{gen});
6.	($Temp_{gen}$):=Mutation(P_m, $Temp_{gen}$);
7.	Pop_{Gen+1}:=Selection($Temp_{gen}$ ∪ Pop_{Gen})
8.	$Gen_{countpso} \leftarrow 0$, assume results are initial population
9.	**while** $Gen_{Count} < Total_{Gen}$
10.	Particlevelocity(Pop_{Gen});
11.	g_{best}:=Assignglobalbest();
12.	UpdatePositions(Pop_{Gen+1})
13.	Increment the $Gen_{countpso}$
14.	Increment the Gen_{Con}
15.	Display gbest as the solution

The algorithm combined the genetic algorithm and particle swarm optimization as described above. Initially, the genetic algorithm operators select the individuals according to their fitness function. Then crossover operator with the single point crossover with probability value Pc =0.8 has been selected. The resulting offspring then go through the mutation with the probability of Pm =0.3. The resultant generation then behaves as the particles with the velocity v, and their velocity changes according to the global best solution. The equations for the change in the position of the particles are given by equation 1.1.

$$v \leftarrow v + c_1 \times rand() \times (Local_{best} - present) + c_2 \times rand() \times (global_{best} - present)$$

The c1 and c2 are the constants with the value of c1, and c2 are assumed to be 2.Equation 2 represents the updates for the position of the present particle. $present \leftarrow present + v$ (1.2)

III. EXPERIMENTAL SETUP

To test the performance of the broker's algorithm developed the four discussed algorithms on cloud report, which at its backend uses the cloud sim for the simulation. For the study, It has been considered three customers who have to generate variableload on each data center. Fig. 1.3 describes the details of customers.

Similarly, the five data centers have been considered and Table 3.1 details thespecification of each of the data centers.

Fig. 1.3: Describing the details of the number of customers and the virtual machines in thesystem

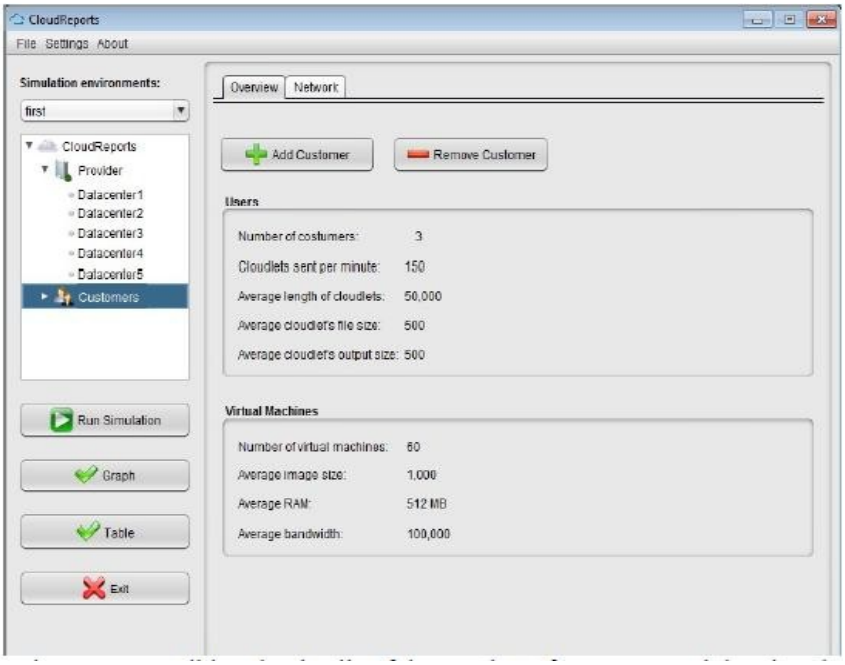

Table 1.1: The Various Specifications of Data Centers

S. No.	Parameters	Values
1	Number of hosts	10
2	Number of processing units	40
3	Processing capacity (MIPS)	96000
4	Storage	20 TB
5	RAM	400 GB

Each data center contains ten hosts, of which 5 use a space-sharing VM Schedulingalgorithm, and the remaining five use the time slice-based VM Scheduling algorithm.

The simulation of the system is done for one hour to observe the performance of the power consumption and the request allocation by the four algorithms.

I. RESULTS AND DISCUSSION

The result obtained is discussed on the three-parameter one by one.

Average Request Completion Time: - Average Request Completion Time is the mean time required by request arriving at the broker from its allocation at the data center and completion. This equation finds Average Request Completion Time as follows:

$$ACT = \frac{\sum Time\ requried\ byeach\ request}{Total\ numerof\ request} \quad (1.3)$$

Fig. 1.4 shows the comparison of the four broker's algorithms. From the figure, it can be observed that the round-robin algorithm is the most inefficient. In contrast, the three algorithms based on the soft computing approach perform better. Still, observingin detail, it can be identified that the Enhance Genetic learning-based Particle Swarm Optimization approach has performed better than the other three techniques.

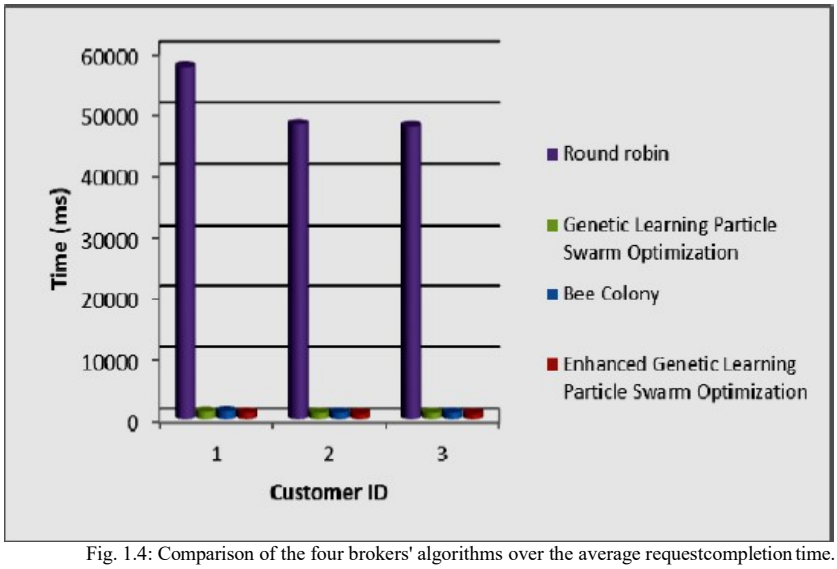

Fig. 1.4: Comparison of the four brokers' algorithms over the average requestcompletion time.

Total Number of Requests Completed: Another comparison of the four broker algorithms has been evaluated using the total number of requests processed by the various data centers. Fig. 1.5 shows that the Bee colony algorithm has performed more efficiently than the EGLPSO algorithm. Still, the EGLPSO has performed better than the other two broker algorithms.

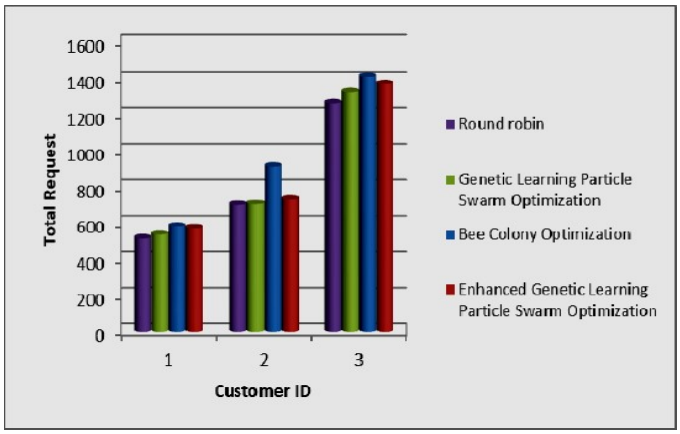

Fig. 1.5: Comparison of the various Broker algorithms by Total number of Request

Table 1.2 details the value obtained for all the four broker policies regarding total andaverage request completion time.

Table 1.2: Details of Total Request Processed and Average Completion Time

Algorithm	Customer	Total Request	Time
Bee Colony Optimization	3	587	1181.577641
	1	918	971.09030786
	2	1417	967.9107757
Enhanced Genetic Learning Particle SwarmOptimization	3	575	957.5802251
	1	736	787.7918144
	2	1375	799.5792987
Genetic Learning Particle SwarmOptimization	3	541	1093.647939
	1	708	898.3115743
	2	1330	908.7170521
Round robin	3	519	57415.65251
	1	703	48090.17949
	2	1273	47712.63415

Powe r Consumption: Fig. 1.5 shows the power consumption of the various broker policies. From the study of the graph shown in figure 7 below, it can be observed that the EGLPSO is consuming more power but also has completed the task earlier. On theother hand, the genetic algorithm has tried to keep the power consumption under control. However, it has taken more time to complete the allocated jobs. The round- robin also took more time to complete the request, which was less than the genetic algorithm and the other broker policies. The running of the machines for more time implies that more cost has to be applied by the user, which will also affect the reputation of the service provider in the market.

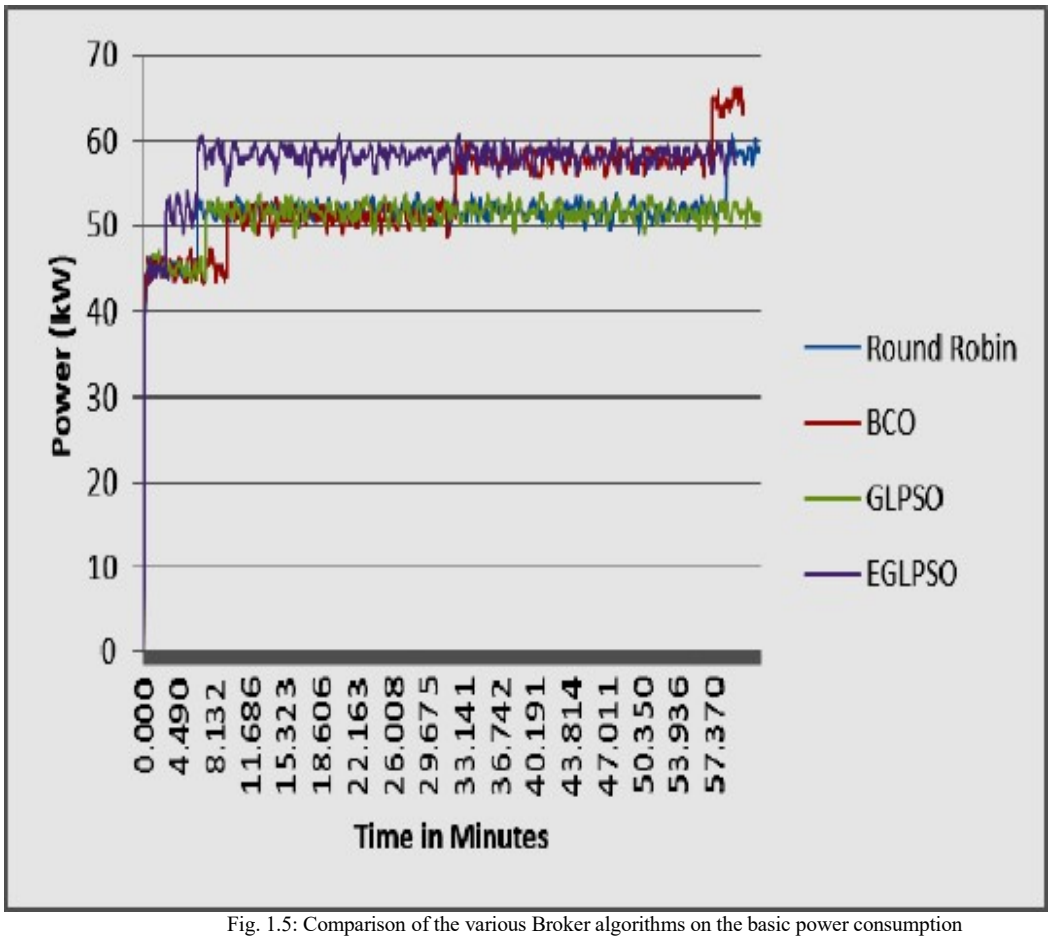

Fig. 1.5: Comparison of the various Broker algorithms on the basic power consumption

II. CONCLUSION

Broker policy is a critical decision factor for the cloud computing environment. The broker allocates the cloudlets to the different data centers during this process. In this paper, It has been considered four strategies for allocating cloudlets to the data center. The broker allocation policies under consideration are round-robin, Bee colony optimization, genetic learning and particle swarm algorithm. Finally, the Enhanced genetic learning-based Particle swarm optimization technique is designed and tested for performance. The Enhanced genetic learning-based particle swarm optimization is the fastest in getting the job done.

In contrast, the round-robin algorithm is the slowest among the four algorithms. The Enhanced genetic learning-based particle swarm optimization technique shows an improvement of 10% over the other soft computing techniques. EGLPSO is 10% faster in comparison to the Bee Colony optimization technique. The Power consumption of EGLPSO is 16% more power than the other broker algorithm. So, this makes the tradeoff between time and power.

REFERENCES

[1] Dam S, Mandal G, Dasgupta K, Dutta P. Genetic algorithm and gravitational emulation based hybrid load balancing strategy in cloud computing. IEEE Third International Conference on Computer, Communication, Control and Information Technology, 2015; 1-7.

[2] Ghumman NS, Kaur, R. Dynamic combination of improved maxmin and ant colony algorithm for load balancing in cloud system. IEEE Sixth International Conference on Computing, Communication and Networking Technologies, 2015; 1-5.

[3] Ramezani F, Lu J, Hussain FK. Task-based system load balancing in cloud computing using particle swarm optimization. International Journal of ParallelProgramming, 2014; 42(5) 739-754.

[4] Cho KM, Tsai PW, Tsai CW, Yang CS. A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing. Neural Computing and Applications, 2015; 26(6) 1297-1309.
[5] Polepally V, & Chatrapati KS. Dragonfly optimization and constraint measure-based load balancing in cloud computing. Cluster Computing, 2019; 1-13.
[6] Arun E, Reji A, Shameem PM, Shaji RS. A novel algorithm for load balancing in mobile cloud networks: Multi-objective optimization approach. Wireless Personal Communications, 2017, 97(2) 3125-3140.
[7] Mishra R, Jaiswal A. Ant colony optimization: A solution of load balancing incloud. International Journal of Web & Semantic Technology (IJWesT) 2012; 3(2) 33-50.
[8] Nishant K, Sharma P, Krishna V, Gupta C, Singh KP, Rastogi R. Load balancing of nodes in cloud using ant colony optimization. Proceedings of IEEE UKSim 14th International Conference on Computer Modelling andSimulation, 2012; 3-8.
[9] Liu Q, Cai W, Shen J, Liu X, Linge N. An adaptive approach to better load balancing in a consumer-centric cloud environment. IEEE Transactions on Consumer Electronics, 2016; 62(3) 243-250.
[10] Zhao J, Yang K, Wei X, Ding Y, Hu, L Xu, G. A heuristic clustering-based task deployment approach for load balancing using Bayes theorem in cloud environment. IEEE Transactions on Parallel and Distributed Systems, 2016; 27(2) 305-316.
[11] Eswaran S, Rajakannu M. Multiservice load balancing with hybrid particle swarm optimization in cloud-based multimedia storage system with QoS provision. Mobile Networks and Applications, 2017; 22(2) 760-770.
[12] Sethi S, Sahu A, Jena SK. Efficient load balancing in cloud computing using fuzzy logic. IOSR Journal of Engineering (IOSRJEN) 2012; 2(7) 65-71.
[13] Lawanyashri M, Balusamy B, Subha S. Energy-aware hybrid fruitfly optimization for load balancing in cloud environments for EHR applications. Informatics in Medicine Unlocked, 2017; 8, 42-50.
[14] Bhatia J, Patel T, Trivedi H, Majmudar V. HTV dynamic load balancing algorithm for virtual machine instances in cloud. Proceedings of IEEE International Symposium on Cloud and Services Computing, 2012; 15-20.
[15] Panwar R, Mallick B. Load balancing in cloud computing using dynamic load management algorithm. Proceedings of IEEE International Conference onGreen Computing and Internet of Things (ICGCIoT) 2015; 773-778.
[16] Xu G, Pang J, Fu X. A load balancing model based on cloud partitioning for the public cloud. Tsinghua Science and Technology, 2013; 18(1) 34-39.
[17] Rastegarfar H, Rusch LA, Leon-Garcia A. Optical load-balancing tradeoffs in wavelength-routing cloud data centers. IEEE/OSA Journal of Optical Communications and Networking, 2015; 7(4) 286-300.
[18] Chen SL, Chen YY, Kuo SH. CLB: A novel load balancing architecture and algorithm for cloud services. Computers & Electrical Engineering, 2017; 58, 154-160.
[19] Naha RK, Othman M. Cost aware service brokering and performance sentient load balancing algorithms in the cloud. Journal of Network and Computer Applications, 2016; 75, 47-57.
[20] Razzaghzadeh S, Navin AH, Rahmani AM, Hosseinzadeh M. Probabilistic modeling to achieve load balancing in expert clouds. Ad Hoc Networks, 2017;59, 12-23.
[21] Fu X, Chen J, Deng S, Wang J, Zhang L. Layered virtual machine migration algorithm for network resource balancing in cloud computing. Frontiers of Computer Science, 2018; 12(1) 75-85.
[22] Dou W, Xu X, Liu X, Yang LT, Wen Y. A resource coallocation method for load-balance scheduling over big data platforms. Future Generation Computer Systems, 2018; 86, 1064-1075.
[23] Shao X, Jibiki M, Teranishi Y, Nishinaga N. An efficient loadbalancing mechanism for heterogeneous range-queriable cloud storage. Future Generation Computer Systems, 2018; 78, 920-930.
[24] Chen T, Marques AG Giannakis GB. DGLB: Distributed stochastic geographical load balancing over cloud networks. IEEE Transactions on Parallel and Distributed Systems, 2017; 28(7) 1866-1880.
[25] Chaczko Z, Mahadevan V, Aslanzadeh S, Mcdermid C. Availability and load balancing in cloud computing. Proceedings of International Conference on Computer and Software Modeling IPCSIT, 2020; 14, 134-140.