Hybrid SFLA-UBS Algorithm for Optimal Resource Provisioning with Cost Management in Multi-cloud Computing

Muhammad Iftikhar Hussain[1], JingSha He[2], Nafei Zhu[3]*
Fahad Sabah[4], Zulfiqar Ali Zardari[5], Saqib Hussain[6], Fahad Razque[7]
Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China[1, 2, 3, 4, 6, 7]
Department of Information & Communication Technologies
Begum Nusrat Bhutto Women University, Sukkur, Pakistan[5]

***Abstract*—Multi-cloud is a vendor-based heterogeneous cloud paradigm in recent era of computing with dynamic infrastructural deployment. Multi-cloud provides all the essential and on-demand requirements of a virtual environment from various domains under a single service level agreement (SLA). Consumers from multitier domains can access all the available resources placed in a shared pool on service provider's side, as per their requirement. The shared pool of resources creates complexity in assigning the best and suitable resource to a particular virtual instance under the same services provider end. The complexity of resources in terms of accessibility from the various domains, dynamic allocation, security, and quality of services (QoS) raises concerns in the multi-cloud infrastructure. This complexity raise concern relates to optimal provisioning and cost management. In the proposed work a hybrid technique with a shuffled leapfrog algorithm and ubiquitous binary search (SLFA-UBS) to resolve these issues with optimal provisioning, dynamic allocation and better resource selection. The proposed work will help to create a need-based and demand-based resource pool with the appropriate selection of each resource. The proposed model also supports resource optimization with dynamic provisioning, cost-effective solution to achieve QoS in multi-cloud deployment on service provider end.**

Keywords*—*Optimal provisioning; resource allocation; multi-cloud; cost management; QoS and selection of resources

I. Introduction

Multi-Cloud is a blend of cloud paradigms, which provides various on demand services with dynamic and elastic resource allocation in a multitier environment. Multi-cloud offers pay per as you go business model for dynamical allocation of resources from a shared pool. Cloud services work under the signed service level agreement (SLA) and standard of architecture (SOA) between a cloud provider and a broker. Cloud computing has distributed nodes, and the end-user does not know about the structural changes and derails of physical resources, allocated on-demand as per-use based on given SLA [1].

Dynamic Resource allocation strategy (RAS) applied on big clusters and data centers in multi-cloud infrastructure, where the pool of resources shared among multitier environment to cover the need base or on demand utilization. RAS provides elasticity of basic hardware, the basic hardware part consists of Hard Disk, Memory, CPU and related I/O devices. These resources allocated or assigned dynamically on demand to achieve performance, QOS and cost management in multi-cloud architecture. Dynamic resource allocation maintains high-level security and resource provision to avoid the overload of resources and achieve the requirement of green computing with optimization [2], [3]. By using RAS, can be avoided the hotspot (less resources than demand) and cold spot (more resources than demand), also plays a vital role to control the other infrastructure-based parameters like efficacy and cost. RSA provide redundant backup link to survive in case of failure, which is a negative impact of infrastructure design, resources scheduling (by using resource provisioning algorithms) used to maintain backup and SLA requirements [4].

Dynamic nature of multi-cloud provides heterogeneity of resources, services, applications, and server in both Virtual Machines (VM) and Physical Machines (PM). Different algorithms used to merge VM and PM depend on the nature of the environment and dependency in terms of resource allocation from shared pool. These algorithms are based on probability or scheduling of resources. Isolation of resources and scalability achieved by using dynamic nature of multi-cloud with dynamic allocation and provisioning of resources.

In multi-cloud deployment where different cloud paradigms and services models combine under a single umbrella and shared pool of resources have heterogeneous requirement in terms of basic structure, deployment model, access rights, transparency and cost. In this architecture, there are lot of security and allocation concerns in terms of end user's rights transparency, shared pool of resources and services, dynamic and runtime allocation, cost management and efficacy. There are many pre-defined techniques for the allocation and optimization of the resources in hybrid cloud environment e.g. scheduling techniques, feature based allocation, priority allocation, clustering and scaling based algorithms. These techniques are useful to overcome dynamic allocation and provisioning in private or hybrid cloud model but these techniques have lack to overcome issues related to multi-cloud model, resources optimization and cost management. The proposed technique will provide a better way to overcome issues related to dynamic and on demand

*Corresponding Author

resource provisioning with optimization and cost management. The proposed model also helps to avoid security issue related to shared-pool of services in multi-cloud and helpful for the new user to figure out basic concerns to adopt multi-cloud services.

Multi-cloud is an archetype that supports vendors to provide services among various private and public clouds holding any blend of these domains, e.g., heterogeneous cloud vendors, accounts, application, services, deployment, security models, premises, regions, and availability zone. The proposed model designed a hybrid algorithm with a shuffled leapfrog algorithm (SLFA) and Ubiquitous binary search (UBS). The design algorithm is helpful for optimization, energy saving, and QoS. In this research deployed a unique algorithm, e.g., genetic algorithm (GA), SLFA, and UBS. The proposed design model shows better results in terms of efficiency and cost-saving model with QoS.

The research contributions are

- Define a new multi-cloud paradigm under IaaS to achieve optimal resource allocation.
- A hybrid algorithm to design the best selection and allocation of resources.
- Fulfill the need-based and demand-based requirements of a virtual environment in multi-cloud.
- Achieve cost-optimized and efficient solutions in a multi-cloud paradigm using SLFA-UBS.

This paper further contains three sections i) related work ii) proposed work iii) optimal hybridization iv) result and discussion v) conclusion.

II. Related Work

Xiaoqun Yuan et al. [5] proposed a game theory that used math's strategies to draw the possible move and functions using Nash equilibrium. The accuracy of the activities makes more chances to win. Each player strategy is the best response for others. The proposed theory depends on memory channel scheduling concerning time. The resource allocation in proposed model assign by using geo distribution with optimal distribution. Mira Morcos et al [6] proposed a greedy algorithm and Nash equilibrium technique for best resource allocation with exponential time. Numerical analysis technique performed form mobile computing network with for resource allocation and maximum cost saving.

Seyedehmehrnaz Mireslami et al. [7] defined a dynamic resource allocation technique on uncertainty-based algorithms and optimization methods and saving total estimated cost. The uncertainty proposed model provides dynamic resource allocation with efficiency and accuracy in cloud infrastructure and deployment. Prasad Devarasetty [8] proposed need-based resource allocation to implement a genetics algorithm technique with a particular defined budget to achieve QoS. The designed techniques deployed on amazon based cloud with cost limitation and efficient resource allocation.

Lailan M. Haji et al. [9] described various techniques for dynamic resource allocation in a cloud environment to achieve efficiency, accuracy, and QoS. J. Praveenchandar et al. [10] proposed an optimal model for dynamic resource allocation and cost-saving in a cloud environment with minimal power management. The authors used a prediction-based dynamic resource table-updating algorithm in the proposed solution to save optimal power. The proposed technique uses impressive task scheduling and power utilization techniques.

On-time deployment access control separation, direct and indirect trust between grid and cloud computing to avoid overlapping and secrecy of credentials, data, and resources. [11] Open stack used on Linux based Xen and KVM machines to extract there features on basic system level. System analysis engine SAE can investigate core resources like memory usage and optimization without any thread and handler. Discrete firefly algorithm applied to avoid side-channel attacks on shared resources to reduce malicious tenant access, energy consumption, and resource loss on the provider end (DFA-VMP) under IaaS [12]. All existing OS-based attacks work in the same way; separate functionally of content reduce risks [13].

Flora Amato et al. [14] stated data security validation and verification techniques using particular classification algorithms on databases. The thermal function used to classify software and hardware classification. Saurabh Singh et al. [15] defined three tire security models related to embed system architecture to achieve trust and VM migration from one source to another associated strategies.

Gururaj Ramachandra et al. [16] defined three attack vector networks, hardware, and hypervisor to retain the complexity of resource allocation and best utilization in a particular virtual system. S Javanmardi et al. [17] proposed application-scheduling parameters in the cloud with fuzzy logic, genetic algorithms, and clustering techniques to achieve QoS in efficiency and throughput. The proposed work helpful to meet the needs of resource pooling and allocation.

K. Dinesh Kumar et al [18] proposed a resource-provisioning model with cost saving strategies to avoid loss of resources. A perdition method presented to avoid over costing, energy lost in data center, and cloud based environment, the proposed method predicted the basic need and upcoming starvation of resources allocation and managed the load as per define perdition. The described techniques enhanced the accuracy, correlation and utilization based perdition of resources in cloud computing. Xiaolong Xu et al [19] presented a meteorological framework in cloud computing with resource provisioning, flat tolerance and load balancing. In the proposed framework, virtual layer 2 further extracted to define meteorological framework then a non-dominated sorting algorithm applied to get load balancing with flat tolerance. Xiaolong Xu et al [20] proposed an uncertainty-based software define framework form edge computing with balanced resource provision and cost in term of energy consumption. A multi-objective dynamic allocation with balanced scheduling techniques adopted to achieve energy efficient and cost effective dynamic resource provision and allocation in fog computing environment.

Cloud computing is multitier, highly scalable, and pay-per-use-based model over the internet; due to this nature, virus, Trojans, and spy are the types of inherited attacks categorized as i) Cloud malware Injection: inject some malware application, service, or VM based machine in Cloud computing [21]. ii) Metadata spoofing: modify metadata information [22]. Younis A. Younis et al. [23] describe other security-related challenges like SLA Monitoring, management and risk analysis, heterogeneity, virtualization, trust, access control, identity management (IDM), and cross-organization management.

B. Asvija et al. [24] describe hypervisor base vulnerabilities with the use of a designed framework based on CPU, memory, and I/O firework in virtualization. Some attacks justified by the proposed technique, like GPU-based side-channel attacks, I/O channel attaches and shared memory side-channel attacks in a shared pool of resources. There will be risk analysis to analyze the performance and deficiency in deployment and measuring security traits. SVM is used to gather the possible online attack over the cloud nodes on a hypervisor level by using parameters system and network-level utilization. This approach consists of detect, remediate, recover, and defend process steps and will be able to improve 90 percent in malware and DOS detection [25].

III. Proposed Work

In the proposed research, in proposed technique presented a hybrid optimization and classification algorithm for dynamic resource allocation in a multi-cloud paradigm. This hybrid algorithm is a combination of (shuffled leaf frog algorithm) SLFA with (Ubiquitous Binary Search) UBS and pure genetics theories. SLFA analyzes all the available resources from shared pool in the multi-cloud and divides them into small groups. Resources from these small groups are selected on need-based and demand-based. Need-based resources are used to complete the initial requirement of each virtual instance. Demand-based resources are required to complete a specific task on a particular VM instance, where needed a full optimization with cost efficiency. Therefore, can be borrowed the resources to meet the runtime need from VM in the same pool or from shared pool. These borrowed resources will be returned back after completion of the task. The proposed technique ensured the optimal utilization of demand-based resource with minimal cost.

A small group of resources created on factors like efficiency, throughput and cost in terms of energy-efficient mode. However, to find the best and a reliable resource pool in the defined mechanism depends on two different search methods internal (need-based) and external (demand base) search. UBS algorithm is used to achieve the best performance and minimum cost to search from everywhere with efficacy and availability of required resource pool for a specific VM in a multi-cloud environment. The flow chart in "Fig. 1" presented the detailed relationship of SLFA and UBS algorithm to attain optimization.

The particular VM generates resources demand request from a shared pool based on the following equation.

VRp (x) = {vRp1, vRp2, vRp3, . . . ,vRpn}

Where VRp (x) presents the number of requests from any VM and x may be any real number from 1 to n. The request generated in a real time-based scenario, depending on the UBS search algorithm. The diagram "Fig. 1" elaborates the relationship between optimization and classification of the proposed dynamic resource allocation technique.

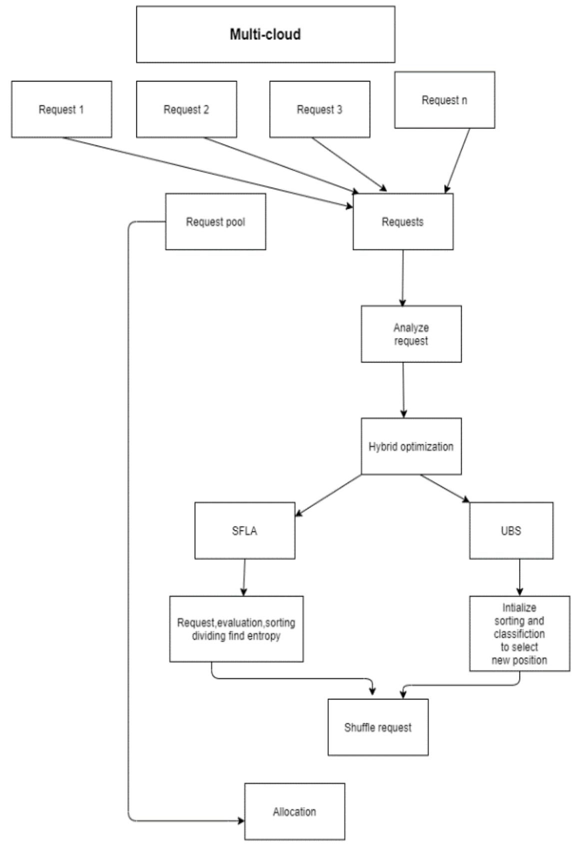


Fig. 1. UBS and SLFA Optimization Framework Flow Chart.

IV. Hybrid Optimization and Classification

Hybrid optimization depends on the deployment of SLFA and UBS in multi-cloud. This combination provides optimal dynamic resource allocation from a shared pool [26]. The hybrid deployment provides the best-optimized solution as compare to prior scheduling and scaling techniques. In this section, this technique presented a deployment of SLFA, UBS, GA, and hybrid SLFA.

A. *Shuffled Leapfrog Algorithm*

In the formation of subgroups, must know about the number of groups and the number of resources in each group. If the number of groups and the number of each type of resources in the group can be considered as one, the total resources will be T = G*R. The number of best utilization of resources depends on the cost function of each group with optimal value. The whole resources divided into small categories; e.g., each type of resource must be in a group to fulfill need-based allocation.

In the division, choose the first member from the first category and second member from the latter category of the resource until the poll completes and add into G. After completion of the division, external search of a particular resource pool will start [27].

B. External Research

The metadata exploring process of the resources divided into two main stages external and internal. This step involves selecting the best optimal pair of resources to complete the initiated request involve the following steps.

Step 1: Initialization: select G and R, where G represents a small group of resources and R the number of resources. So the total number of groups with complete resources will be.

T = G*R

Step 2: Reproduction of Virtual groups: For the available resources, the sample of S virtual resources will be v(1), v(2) …. V(s).

V(i) = {vi_1, vi_2, …, vi_d} where d present the matter of decision and selection of particular resource in the group.

Step 3: Section and sorting: Sort the resources in descending order in a particular group. Then complete group will be v(i), s(i) and i = 1 … to s. the best place of the resources is (v =Px(i)).

Step 4: Division of resources into groups: Divide array X into Y that each of them obtains N resources.

Step 5: Resource evaluation in each group: Each Yk where k = 1,2,3 … G in each group assessed by the internal search given bellow.

Step 6: Combination of groups: After the evaluation, each group contains (Y1 … YG) the number of each type of resource for a specific pool.

Step 7: If the convergence conditions meet, then stop else; go to the fourth step of external research.

C. Internal Research

In the 5th step of the external search, the selection of group performed N time independently. After the completion of the search mechanism, the algorithm will return to global research to complete the step. Internal research steps are as follows.

Step 1: Set iG and iN to zero. Where iG counts the number of groups and iN counts the progression steps.

Step 2: 1 + iG = iG

Step 3: 1 + iN =iN

Step 4: Creation of subgroups: The creation of subgroups related to higher value associated with best one and lower value associated with a lower one. Value is assigned with the help of triangular probability distribution.

Step 5: Correction of the worst position: This is calculated by combining the lowest associated probability and selection leap parameter if the resource e.g., Ram, has the lowest capacity than the needed one. Now, if reassigned the value to a particular resource, then in the best case, go for step 8 of the external search; otherwise, go to step 6 of the external searches.

Step 6: calculate the resource size with the best-assigned value. If results obtained from step 5 are not better, then the size of the particular resource can be calculated as efficiency. After reassigning value to a particular resource, if its efficiency is better than the previous one, it can be replaced with the previous one's value. Otherwise, go to the next step of internal research.

Step 7: If the new value of the resources is not according to the given need, then randomly generate a new virtual resource and replace that particular resource with a randomly generated new virtual resource in the pool.

Step 8: Update the group: After changing the value of the lowest resource with the newly virtually generated resource, sort them in descending order with the UBS algorithm's help.

Step 9: if N > iN perform step three of the internal search again.

Step 10: if G > iG goes to step 1 of internal search, otherwise return to the external search to combine the groups.

D. Ubiquitous Binary Search

UBS is all aware of the binary search algorithm, very complex to resolve the required query. Binary search algorithm always follows a sequential search pattern, so it is very costly and complex to get output. The cost of the algorithm is highly concerning the complexity of the search scenario. UBS can search everywhere, anytime, with any sequence from any medium or device. Hence, it is very helpful to implement in current distributed nature infrastructure like multi-cloud. In our proposed model, UBS used with no loops and no equal check. Like can find the required output by using <= and >=, which is very helpful to reduce review and loops and increase the efficiency of the algorithm with low cost. The low and high method or use to select with an appropriate resource for the best selection in low will always less than high and a mid to calculate the average. The cost of the algorithm will be lg(n).

```
def ubiquitous_binary_search(a,key) # a is the array and key is the value we want to search

    lo= 0
    hi = a.length-1

    while(hi-lo>1)
     mid = lo + (hi-lo)/2

     if a[mid]<=key
        lo=mid
     else
        hi=mid
     end
    end

    if (a[lo]== key)
       return lo
    else
       return "value not found"
    end
end
```

E. Genetic Algorithms

Genetic algorithms are used in natural section techniques to sort out complex problems. These algorithms define a given problem as a string in the workspace and find the result with core gentian parameters in that give model.

GA algorithms are probabilistic and use natural evaluation in the propagation of results. The population of biological metrics like chromosomes used as an idle parameter to emulate GA algorithms. As the different types of resources in a resource pool, select and the best match one to overcome demand-based and need base requirements of a particular system in a multi-cloud architecture. Nature resources parameters are used to emulate and propagate the best results in the selection pool.

F. Hybrid SFLA and UBS

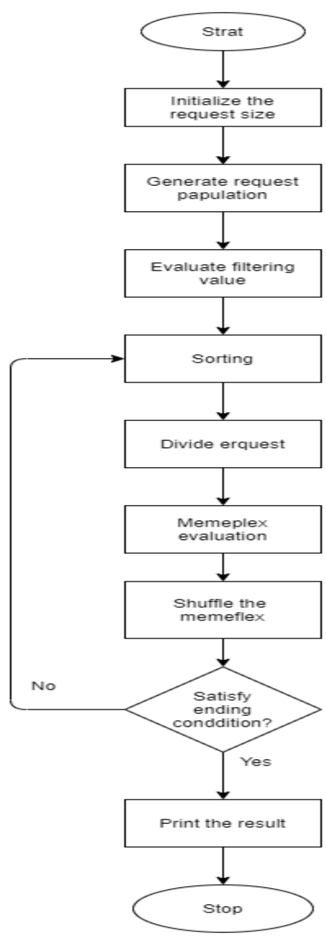
In the hybrid proposed solution, SFLA used with UBS qualities together and founded better results. The proposed hybrid algorithms showed more throughput, efficiency, and cost savings in terms of execution time and turnaround time. UBS used as a searching algorithm to find the best resource for selection and get a better combination of need-based and demand-based requirements. The proposed model helpful to choose the best appropriate resource to fit in the required need of a specific VM in multi-cloud infrastructure deployment with hybrid vendor-based architecture.

The proposed model described in the form of flow chart-based implementation in “Fig. 2”.

The cost of this hybrid model in term of execution time and turnaround time calculated as

- Worst case time: Ø lg(n).
- Average case time: Ø (1).
- Average case time: Ø lg(n).

Therefore, gain efficacy and energy-saving model by using the proposed hybrid algorithm for dynamic resource allocation in multi-cloud.



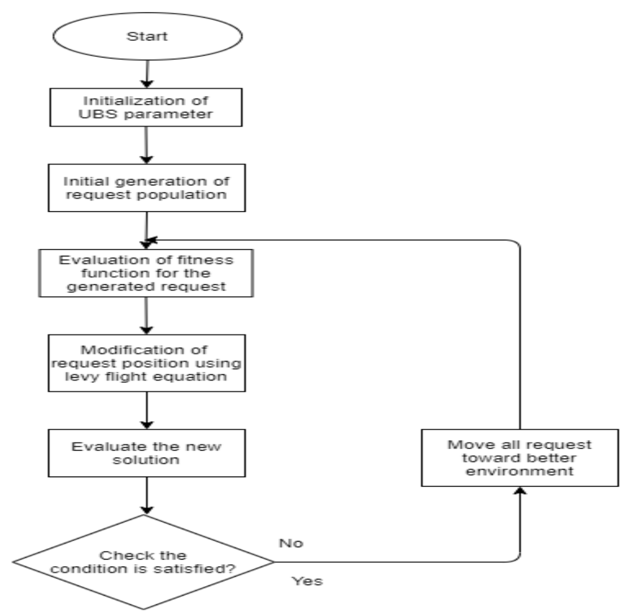


Fig. 2. Hybrid SLFA and UBS Proposed Model Flow Chart.

V. Results and Discussion

In the proposed model throughput, Turnaround time, and execution time as metrics to evaluate the QoS and cost of the design algorithm.

A. Throughput

Throughput is maximum output at a certain time for a specific hardware type. It is performance measure of basic resources such as RAM, Hard drive, and CPU in a shared resources pool [28]. The throughput calculated as

$$T_t = I_t/t$$

According to the statistics shown below diagram, T_t presents the maximum output of a certain resource at a specific time. I_t is the peak value of resource under time unit "t." Results shown that the proposed hybrid algorithm present maximum throughput, which helps to create and maintain a complete resource group according to the given need. The throughput values taken at the point where maximum number of task executed by each algorithm individual. In designed case, the maximum number of tasks up to 300. The graphical presentation of different algorithms with respect to times per second is illustrated in "Fig. 3". Where "Table I" presented the numeric values of throughput in terms of per second.

B. Turnaround Time

Turnaround time presents the maximum amount of time from submitting a specific task, and its output returns to the user. It totally depends upon the function used by the developer in the code. Hence, UBS performed with less turnaround and execution time due to the absence of loop complexity in generic binary algorithms [29]. It can be calculated as

$$T(t)avg = C(t) - A(t)$$

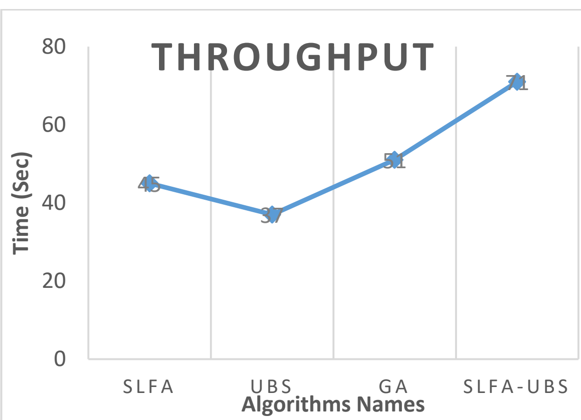


Fig. 3. Throughput Graph.

TABLE I. Throughput on a Certain Define Point with Maximum Number of Tasks 300

Name	Time (Sec)
SLFA	45
UBS	37
GA	51
SLFA-UBS	71

Where T(t)avg presented average turnaround time with CT, total execution time and A(t) time of arrival of a specific task. "Table II" described the turnaround time as per number of tasks for each algorithm.

Results showed that the hybrid SLFA-UBS algorithm performs continually constant with the increase of the number of tasks on the horizontal axis. The proposed model is much efficient to resolve maximum number of tasks in less turnaround time as shown in the "Fig. 4".

C. Execution Time

Total time requires executing a certain task as per user's demand. The execution time of SLFA-UBS is less, as compare to individual performances of every algorithm shown in "Table III" and "fig 5" respectively. Execution time can be calculated as

$$E(T) = E(t) - F(t)$$

Where E(T) presents the computational time required to execute a specific task, E(t) ending of the task, and F(t) is the beginning of a specific task by the user. With respect to the number of tasks over total time to execute, SLFA-UBS performs better than any individual algorithm.

TABLE II. Turn Around Time Comparison

Number of tasks	SLFA	UBS	GA	SLFA-UBS
50	21	25	35	13
100	23	27	37	13
150	24	28	39	15
200	25	30	40	15
250	26	31	41	17
300	27	32	42	17

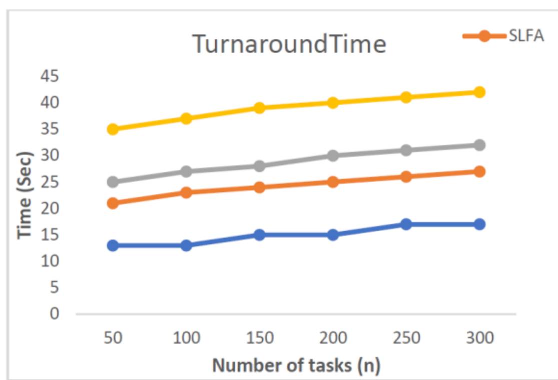


Fig. 4. Turnaround Time Graph.

TABLE III. Execution Time Comparison

Number of tasks	SLFA	UBS	GA	SLFA-UBS
50	19	17	15	7
100	21	23	25	12
150	27	25	23	14
200	31	32	35	15
250	34	35	38	18
300	37	39	44	20

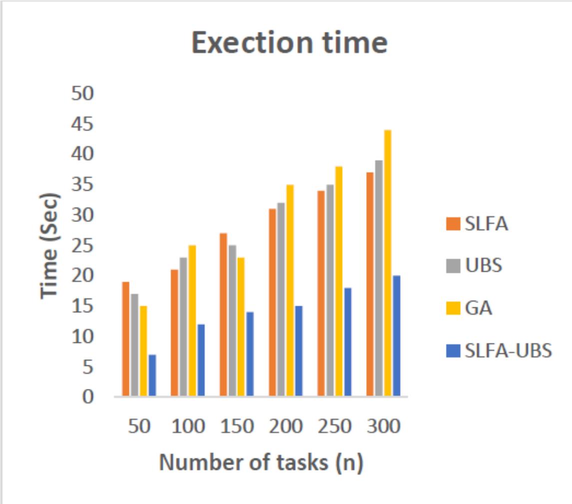


Fig. 5. Execution Time Graph.

VI. Conclusion

In the multi-cloud deployment dynamic resource allocation and best resource selection for particular virtual machine to fulfill the need-based resources requirement is a big challenge, and creates an adequate concern for cloud services providers. Resource optimization with minimal computational cost and throughput is also a rising concern with respect to better resource allocation in multi-cloud environment. SLFA-UBS proposed algorithm used to resolve the problems related to optimization and cost management in the selection of particular need-based resources from a shared pool in a multi-cloud paradigm. Execution of the proposed model used "CloudSim" to simulate and extract the results. The results showed that the throughput of the proposed model is 60% better than a single individual algorithm. The turnaround time and execution time had very few seconds variations near to constant as per increase in the number of tasks.

The proposed work ensured resources optimization with the dynamic provisioning techniques in the multi-cloud environment. Cost efficiency and quality of services also obtained in the summarization of this research work. Hence, we concluded that our proposed SLFA-UBS algorithm obtained better performance in optimal dynamic resource provisioning with QoS and low cost. In the future we extend our work to real time deployment of designed algorithm on cloud service providers end to enhance the archived results in test environment.

References

[1] Kumar, R., & Goyal, R. (2019). On cloud security requirements, threats, vulnerabilities and countermeasures: A survey. Computer Science Review, 33, 1-48.

[2] Diouani, S., & Medromi, H. (2018). Green cloud computing: Efficient energy-aware and dynamic resources management in data centers. International Journal of Advanced Computer Science and Applications, 9(7), 124-127.

[3] Ahmad, I., & Chang, K. (2020). Mission-critical user priority-based cooperative resource allocation schemes for multi-layer next-generation public safety networks. Physical Communication, 38, 100926.

[4] Suresh, A., & Varatharajan, R. (2019). Competent resource provisioning and distribution techniques for cloud computing environment. Cluster Computing, 22(5), 11039-11046.

[5] Yuan, X., Min, G., Yang, L. T., Ding, Y., & Fang, Q. (2017). A game theory-based dynamic resource allocation strategy in geo-distributed datacenter clouds. *Future Generation Computer Systems*, *76*, 63-72

[6] Morcos, M., Chahed, T., Chen, L., Elias, J., & Martignon, F. (2018). A two-level auction for resource allocation in multi-tenant C-RAN. *Computer Networks*, *135*, 240-252.

[7] Mireslami, S., Rakai, L., Wang, M., & Far, B. H. (2019). Dynamic cloud resource allocation considering demand uncertainty. *IEEE Transactions on Cloud Computing*.

[8] Devarasetty, P., & Reddy, S. (2019). Genetic algorithm for quality of service based resource allocation in cloud computing. *Evolutionary Intelligence*, 1-7.

[9] Haji, L. M., Zeebaree, S. R., Ahmed, O. M., Sallow, A. B., Jacksi, K., & Zeabri, R. R. (2020). Dynamic resource allocation for distributed systems and cloud computing. *TEST Eng. Manag*, *83*, 22417-22426.

[10] Praveenchandar, J., & Tamilarasi, A. (2020). Dynamic resource allocation with optimized task scheduling and improved power management in cloud computing. *Journal of Ambient Intelligence and Humanized Computing*, 1-13.

[11] Kaur, S., & Bhushan, R. (2013). Review Paper on Resource Optimization of Servers using Virtualization. *International Journal of Advance Research in Computer Science and Software Engineering*, *3*, 327-332.

[12] Ding, W., Gu, C., Luo, F., Chang, Y., Rugwiro, U., Li, X., & Wen, G. (2018). DFA-VMP: An efficient and secure virtual machine placement strategy under cloud environment. *Peer-to-Peer Networking and Applications*, *11*(2), 318-333.

[13] Mohd Hairy Mohamaddiah, Azizol Abdullah, Shamala Subramaniam, Masnida Hussin "A Survey on Resource Allocation and Monitoring in Cloud Computing", International Journal of Machine Learning and Computing, Vol. 4, No. 1, February 2014.

[14] Amato, F., Moscato, F., Moscato, V., & Colace, F. (2018). Improving security in cloud by formal modeling of IaaS resources. *Future Generation Computer Systems*, *87*, 754-764.

[15] Singh, S., Jeong, Y. S., & Park, J. H. (2016). A survey on cloud computing security: Issues, threats, and solutions. *Journal of Network and Computer Applications*, *75*, 200-222.

[16] Ramachandra, G., Iftikhar, M., & Khan, F. A. (2017). A comprehensive survey on security in cloud computing. *Procedia Computer Science*, *110*, 465-472.

[17] Javanmardi, S., Shojafar, M., Persico, V., & Pescapè, A. (2020). FPFTS: A joint fuzzy particle swarm optimization mobility-aware approach to fog task scheduling algorithm for Internet of Things devices. Software: Practice and Experience.

[18] Kumar, K. D., & Umamaheswari, E. (2018). Prediction methods for effective resource provisioning in cloud computing: A survey. Multiagent and Grid Systems, 14(3), 283-305.

[19] Xu, X., Mo, R., Dai, F., Lin, W., Wan, S., & Dou, W. (2019). Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud. IEEE Transactions on Industrial Informatics, 16(9), 6172-6181.

[20] Xu, X., Cao, H., Geng, Q., Liu, X., Dai, F., & Wang, C. (2020). Dynamic resource provisioning for workflow scheduling under uncertainty in edge computing environment. Concurrency and Computation: Practice and Experience, e5674.

[21] Jensen, M., Schwenk, J., Gruschka, N., & Iacono, L. L. (2009, September). On technical security issues in cloud computing. In *2009 IEEE International Conference on Cloud Computing* (pp. 109-116). Ieee.

[22] Jensen, M., Gruschka, N., & Herkenhöner, R. (2009). A survey of attacks on web services. *Computer Science-Research and Development*, *24*(4), 185.

[23] Younis, Y. A., & Kifayat, K. (2013). Secure cloud computing for critical infrastructure: A survey. *Liverpool John Moores University, United Kingdom, Tech. Rep*, 599-610.

[24] Asvija, B., Eswari, R., & Bijoy, M. B. (2019). Security in hardware assisted virtualization for cloud computing—State of the art issues and challenges. *Computer Networks*, *151*, 68-92.

[25] Veloudis, S., Paraskakis, I., Petsos, C., Verginadis, Y., Patiniotakis, I., Gouvas, P., & Mentzas, G. (2019). Achieving security-by-design through ontology-driven attribute-based access control in cloud environments. *Future Generation Computer Systems*, *93*, 373-391.

[26] Brabra, H. (2020). *Supporting management and orchestration of cloud resources in a multi-cloud environment* (Doctoral dissertation, Institut Polytechnique de Paris; Université de Sfax (Tunisie). Faculté des Sciences économiques et de gestion).

[27] Misener, P. (2020). *Food Insecurity and College Athletes: A Study on Food Insecurity/Hunger among Division III Athletes* (Doctoral dissertation, State University of New York at Binghamton).

[28] Kwan, A., Wong, J., Jacobsen, H. A., & Muthusamy, V. (2019, July). Hyscale: Hybrid and network scaling of dockerized microservices in cloud data centres. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (pp. 80-90). IEEE.

[29] Lin, W., Peng, G., Bian, X., Xu, S., Chang, V., & Li, Y. (2019). Scheduling algorithms for heterogeneous cloud environment: main resource load balancing algorithm and time balancing algorithm. *Journal of Grid Computing*, *17*(4), 699-726.