Library Progress International
Vol.44, No.1s, January-June 2024: P.115-121
DOI: https://doi.org/10.52710/lpi.44.1s.10

Print version ISSN 0970 1052
Online version ISSN 2320 317X

Original Article

Available online at www.bpasjournals.com

SLA-Based Resource Provisioning in Library Systems Utilizing Genetic Algorithm

[1]Meenakshi Saini[*] and [2]Neeraj Mangla

Author's Affiliation:
[1 2]CSE Department, Maharishi Markandeshwar Engineering College
Maharishi Markandeshwar (Deemed to be University) Mullana, Ambala, Haryana, India

How to cite this article: Meenakshi Saini, Neeraj Mangla (2024). SLA-Based Resource Provisioning in Library Systems Utilizing Genetic Algorithm. *Library Progress International,* 44(1s), 115-121.

ABSTRACT
In the context of library science, cloud computing represents a cutting-edge approach, enabling libraries to dynamically share computational resources as needed. This technology leverages underutilized computational power within a networked environment to provide seamless services to users. To ensure a higher Quality of Service (QoS), libraries and Cloud Service Providers (CSPs) establish Service Level Agreements (SLAs). However, increasing user demand often prevents CSPs from maintaining the agreed QoS, resulting in SLA breaches. This research article presents an algorithm for efficient dynamic resource provisioning, addressing various SLA-related penalties. By incorporating a Genetic Algorithm (GA) into the proposed approach, the algorithm effectively schedules user requests and optimizes the use of Cloud-based servers. It also includes an SLA negotiation method to reduce penalty costs and minimize SLA breaches, ensuring reliable and efficient library services.

KEYWORDS
Cloud Service Provider, Service Level Agreement, Quality of Service**,** Virtual Machine.

1. Introduction

Businesses and research facilities have started using on-demand Cloud services to outsource their information technology and computational needs as Cloud computing gains popularity [1]. Datacenters that are extensively virtualized housing tens of thousands of computers often make up the Clouds. Although these virtualized infrastructures have many benefits, such as the capacity to scale resources on demand; there are still problems that hinder their broad use in Clouds. In particular, the Cloud datacenters must offer greater and more stringent QoS assurances in order for this computing paradigm to be commercially successful. These assurances that are listed in SLA are the only way that can feel consumers about the security of their work on the Cloud [2]. Resource provisioning is essential to make sure that CSPs fulfill their promises to clients in a way that maximizes the use of underlying infrastructure. An effective resource management plan would automatically assign to each service request with minimum resources required for satisfactory SLA fulfillment, thus, freeing up extra resources to build more Virtual Machines (VMs). The provisioning decisions must be flexible enough to accommodate changes in load as they happen and elegantly handle unexpected spikes in demand. Because of these factors, automatically dividing datacenter resources across the numerous hosted apps is a difficult operation. Currently, a greater variety of applications with various SLA needs are hosted by Cloud datacenters [3]. The fundamental disparities between these various workloads add to the difficulty of the resource supply operation

[4]. First, various applications have varied SLA needs. Response time and throughput guarantees are necessary for transactional applications, whereas performance issues with non-interactive batch jobs exist. Second, batch operation's resource demands may be foreseen to a greater extent than those of transactional applications, such as Web applications, which have a tendency to be very unpredictable and busy in nature [5]. Therefore, to accommodate worst-case demand, excessive provisioning of resources has been employed in order to satisfy SLA requirements. However, during non-peak hours, servers are often used at extremely low levels results in the wastage of resources. Additional maintenance expenses, such as those associated with server cooling and management, emerge from this over-provisioning of resources [6]. Many researchers attempted to overcome these challenges through dynamic provisioning of resources. The research is currently in its early stages and involves a variety of application types and SLAs, despite the fact that computationally intensive apps are progressively becoming a component of business datacenters. The majority of datacenters nowadays run several application types on multiple VMs without being aware informed of their specific SLA requirements, which may lead to resource under-utilization and administrative overhead [7].

2. Related Work

Quiroz et al. [3] describe a decentralized, reliable online clustering technique. It does not take into account the SLA penalty and employs a fixed number of VMs.

Carrera et al. [5] offers a method to successfully handle heterogeneous tasks, including batch processes and transactional applications. The main objective of this study was to utilize datacenter resources as effectively as possible while meeting the apps diverse SLA needs.

Zhang et al. [8] developed a method using ghost VMs to promptly redistribute the resources for a virtualized utility-based computing platform. The work presented by the author focused on a constant number of VMs.

Wang et al. [9] evaluated the additional burden associated with a dynamic allocation method over a static allocation strategy in terms of system capacity and application-level performance. In this study, the concept of dynamic allocation is broadened to include Web and high performance computing workloads.

Aljoumah et al. [10] provides research on the overall structure of SLA, its elements, management techniques, SLA lifetime, and pricing.

de Asís López-Fuentes et al. [11] discussed that Cloud computing still has to overcome a number of obstacles in the areas of heterogeneity, collaboration, privacy, security, resilience, throughput, and scalability. These difficulties will result in the introduction of fresh holistic designs, teamwork tactics, and distribution infrastructures. Peer-to-Peer networks and distributed infrastructures have become prominent as management-related solutions.

Panda et al. [12] provides two SLA-based work scheduling methods for a heterogeneous multi-Cloud environment, namely SLA-MCT and SLA-Min-Min. The first approach uses a single step of scheduling, whereas the second algorithm uses two phases.

Hussain et al. [13] offers SLA-RALBA, a cost-effective SLA-based load balancing scheduler, for heterogeneous Cloud infrastructures. The suggested method accommodates the three levels of SLA selected by Cloud users.

According to Wang et al. [14], in order to enable Cloud storage more effectively, suggest a resource scheduling algorithm that is mindful of SLAs. The suggested method simultaneously maximizes I/O throughput and back-end node space utilization.

Saini et al. [15] evaluated a Multi-Objective Genetic Algorithm (MOGA) on cost and makespan. The suggested MOGA gives better outcomes to schedule tasks on the processors and minimizes the budget and makespan.

Saini et al. [16] analyze the efficiency of various meta-heuristic algorithms in resolving the workflow scheduling difficulty in the Cloud environment. Meta-Heuristic scheme is one such method to achieve the best or close to perfect planning of undertaking arrangement for the Cloud situation.

Mangalampalli et al. [17] presents a task-scheduling technique that takes VMs and task

priority into account. In order to simulate this scheduling paradigm, author had opted for Whale optimization.

Lan et al. [18] provide a technique for optimization to improve SLA assurance efficiency during request dispatch. SLA-ORECS's examination demonstrates notable performance improvements, especially with regard to average time consumption and system throughput.

3. Research Metodology

3.1 SLA-based Resource Management Framework

At first, the Cloud user must register with the resource broker using the online portal by completing the registration requirements in order to enjoy the services supplied by the service supplier. The user might submit a request to the service broker after completing the process of registration. Requests from users include information regarding VMs configuration, required VM quantity, operating system type, required software, required number of days, etc. The service broker receives this user request data. When a consumer requests a service, the broker will determine the resources that are available, the anticipated processing time and cost of providing the benefits. The supplier of the service is then given this information. The agent looks for another supplier of services to fulfill the consumer's demand if the service supplier is unable to do so. After the service request has been fulfilled, SLA describing the rules and regulations that must be met establishes and provides resources between both clients and CSPs occurs. Enhanced customer happiness, improved service quality, establishing and preserving a good working connection between the consumers and the suppliers are benefits of this approach.

3.2 SLA Manager

- The SLA manager's job is to ensure that all SLA for different client service requests are fulfilled in accordance with requirements. The SLA manager is in charge of monitoring the service provider's and customer's compliance with SLAs. SLA manager is made up of elements like:
- **Service Request Examiner (SRE):** The use of Web apps from anywhere in the globe, SRE submits service request submissions on behalf of users. After receiving a service request, SRE first verifies the consumer's identity and assesses their QoS requests before deciding whether to grant or deny the request.
- **SLA Generator:** SLA generator formalizes a SLA that guarantees the human rights of the clients and relates to their conditions, bridging the gap between SLA reality and QoS. Here, an agreement exists between the buyer and the CSP and is signed. It contains service standard goals in accordance with the QoS requirements laid out between the client and the CSP.
- **SLA Negotiator:** After helping customers successfully negotiate the best deal with several service providers, SLA negotiator will commend them for successfully obtaining the necessary service. If an SLA violation occurs during service provisioning, a penalty must be applied.

3.3 SLA Violation and Negotiation

Many factors, including alterations in the environment, software faults, network performance, bandwidth, and others, may have an impact on how the system behaves in a Cloud computing environment. The most prevalent crucial elements of Cloud computing is SLA breach, which lowers user satisfaction levels while also upsetting CSPs and may result in punishment or fines. SLA violations can occur in a variety of situations, including when services are not given at all, performance is provided below the agreed-upon level, services are provided at the proper level but with extra delay, and services are used differently than expected when it comes to VM resources. In order to use services, SLAs must be negotiated. The mechanism for negotiation determines the relative importance of both the parties (consumers and service providers), their roles in the negotiation, the visibility of the deals made, how the session will be run, the constraints of the negotiation process, etc. SLA negotiation is a crucial technique to guarantee Cloud service efficiency and boost trust between Cloud service users and CSPs. It is possible to outline the QoS demands of crucial

service-based activities through a SLA agreement between Cloud parties.

4. Proposed Work

The implementation of SLA based Resource Management Framework (SLARMF) focuses on various forms of SLAs in addition to maximizing resource consumption. This approach improves the connection among the CSP and the client by reducing the number of breaches of SLAs while providing a guaranteed QoS to the customers as defined in the SLA.

4.1 Data Sets

We used workload information for transactional apps for our tests. Transactional data is gathered through:

- Planet Lab monitoring infrastructure. The information includes the processor, storage, and speed use of over a thousand machines spread across around 500 locations worldwide (https://codeen.cs.princeton.edu/comon/).
- NEC Personal Cloud Trace. The storage layer and sharing interactions are two sources of data that are combined in the NEC dataset (https://Cloudspaces.eu/results/datasets, https://sites.cs.ucsb.edu/~rich/workload/).

4.2 Assumptions

- $VM_{cost} = CSP_{initialcost} * (Required\_RAM\_Size/0.5)$
- Total_ RAM_Cost = RAM_Cost * Number_of_days
- Total_VM_Cost = Total_RAM_Cost * Number_of_VMs
- Where
- $CSP_{initialcost}$ is CSP's RAM initial cost.
- Required_RAM_Size is Cloud user requested RAM Size.
- Total_ RAM_Cost is Single VM Cost.
- Number_of_VMs is a number of VMs needed by the Cloud users.
- Total_VM_Cost is Total VM Cost for a Cloud user.
- Penalty cost is computed as follows:
- Single_VM_Cost = Total_VM_Cost/ Number_of_VMs
- Where
- Single_VM_Cost is Single VM Cost.
- Total_VM_Cost is Total Cost of VM purchased by a Cloud user.
- Number_of_VMs are number of VMs purchased.
- Minimum_Penalty_Cost = (Single_VM_Cost*SLA/100)
- Where
- Minimum_Penalty_Cost is Minimum Penalty Cost.
- SLA's worth will change based on the service provider.
- Penalty_Cost= ((Minimum_Penalty_Cost + Single_VM_Cost)*RVM)
- Where
- Penalty_Cost is Penalty Cost.
- RVM stands for "VMs Cancelled by CSP."

4.3 Mathematical proof

Fitness Calculation:

$$F = (VM_{request\_time}, VM_{response\_time})(1)$$

Finding the Best Fitness:

$$F_i = \min(F)(2)$$

Iteration Loop:

$$\text{while } T \leq T_{max}(3)$$

New Solution Fitness

$$F_j = f_{new}(VM_{request\_time}, VM_{response\_time})(4)$$

Replace with Better Solution:

$$\text{if } F_i > F_j \text{ then } F_i = F_j(5)$$

Worst Solution Replacement:

$$F_{worst} = \max(F)(6)$$

$$\text{Replace } F_{worst} \text{ with a new solution}$$

Ranking and Finding Best Solution:

$$F_{best} = \min(F)(7)$$

4.4 Algorithm: SLARMF-GA

- *Step 1: Obtain total number of Data Centre's (DCn)*
- *Step 2: While (DCn <> NULL) do*
- *Cluster the VMs requests based on Client's submissions*
- *Step 3: Generate CU, CSP and F for each clustered Hosts (Hostc)*
- *CU= Cloud User request time (VMrequest_time)*
- *CSP = CSP response time (VMrespone_time)*
- *F = Calculate fitness (VMrequest_time, VMrespone_time)*
- *Step 4: Find the best fitness value (Fi) from F*

- *While (T <= Maximum iteration)*
- *Compute new solution (Fj)*
- *If (Fi> Fj) then the replace the new solution by j*
- *End if*
- *End While*
- *Worst solution is abandoned and replaced by new solution*
- *Provide ranking to the solution and find the current best solution*
- *End While*

5. Results and Discussion

5.1. Case 1: SLA% Non-Zero

Relevant to users of the Cloud, CSP offers VM. CSP has cancelled a few VMs that Cloud Users have already purchased. Penalty costs must be paid by CSP depending on SLA.

5.1. Case 2: SLA% is Zero

Cloud Users are receiving VM from the relevant CSP. No VM that Cloud Users have already purchased is being cancelled by CSP. CSP is exempted from paying penalties based on SLA.

5.1. Case 3: SLA% is NA

The CSP in this case is not offering the users complete services. As a result, the CSP hasn't made any money and is therefore exempt from paying a fine.

Table 1. Penalty cost computation default SLA vs. SLARMF-GA (Planet Lab Workload Data)

Cloud Service Providers	SLA Framework (Cloudsim Default) ($)	SLARMF-GA ($) (Proposed)
CSP-1	200	175
CSP-2	0	0
CSP-3	410	380
CSP-4	0	0
CSP-5	580	560
CSP-6	0	0
CSP-7	0	0
CSP-8	1310	1275
CSP-9	0	0
CSP-10	0	0

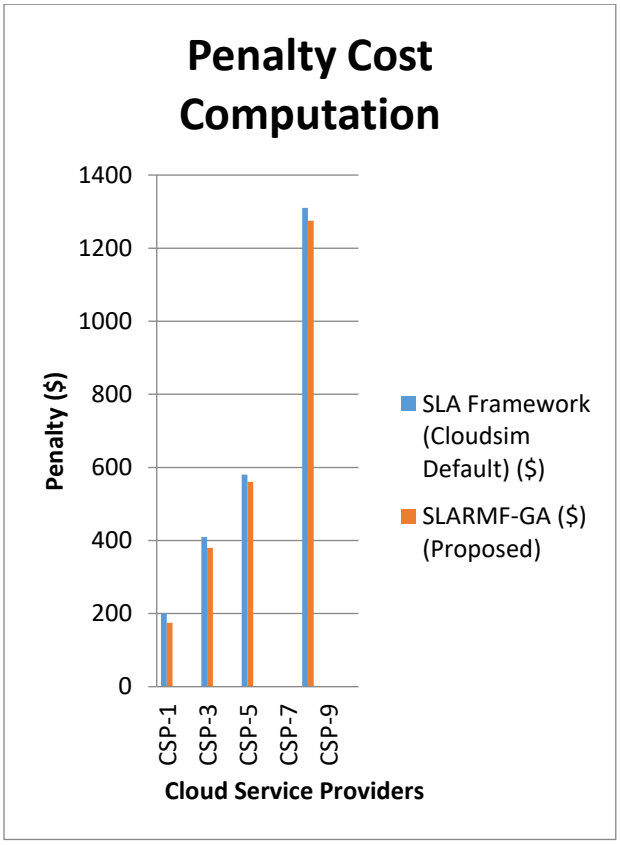

Fig. 1. Penalty cost computation default SLA vs. SLARMF-GA (Planet Lab Workload Data)

Table 1 & figure 1 compares the penalty cost computation for Planet Lab Workload Data using the SLA-based model versus the proposed technique (SLARMF-GA) for all service providers. Here, the services supplied by CSP-2, CSP-4, CSP-6, CSP-7, CSP- 9 and CSP-10 service providers without any breaches in both the cases. Further, the proposed technique (SLARMF-GA) shows less penalty in comparison with default SLA.

Table 2. Penalty cost computation default SLA vs. SLARMF-GA (NEC Workload Data)

Cloud Service Providers	SLA Framework (Cloudsim Default) ($)	SLARMF-GA($)
CSP-1	189	175
CSP-2	230	198
CSP-3	0	0
CSP-4	670	590
CSP-5	0	0
CSP-6	890	810
CSP-7	345	310

CSP-8	0	0
CSP-9	789	678
CSP-10	456	390

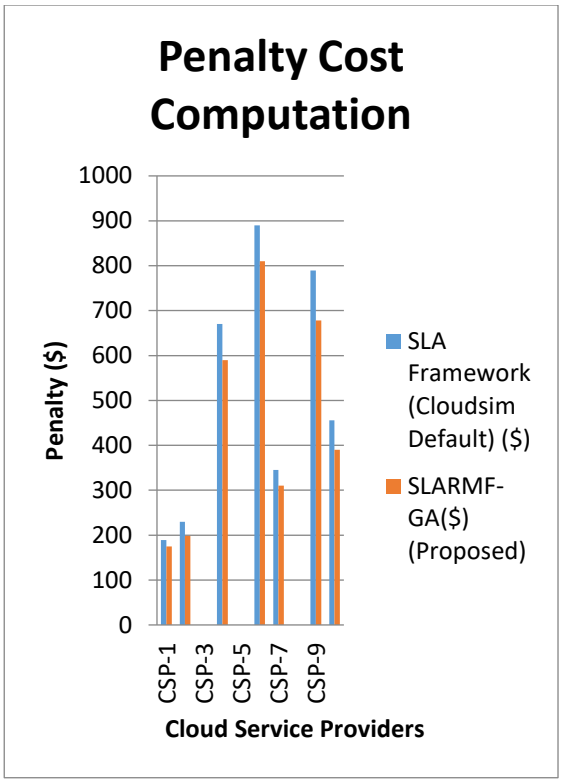

Fig. 2. Penalty cost computation default SLA vs. SLARMF-GA (NEC Workload Data)

Table 2 & figure 2 compares the penalty cost computation for NEC Workload Data using the SLA-based model versus the proposed technique (SLARMF-GA) for all service providers. The service providers from CSP-3, CSP-5, and CSP-8 here supply the services without any breaches in both the cases. Further, the proposed technique (SLARMF-GA) shows less penalty in comparison with default SLA.

6. Conclusion and Future Scope

In addressing the dynamic needs of library system, we simulate the suggested SLA-based resource management system. By decreasing SLA violations, the suggested SLA-based resource management system enhances the bond between both the client and the CSP. The proposed SLA-based resource management system makes optimal utilization of available resources and ensuring that customers receive the level of service specified in the SLA. The suggested approach can reduce breaches of SLAs and unsuccessful negotiations while improving cost efficiency. In order to suggest the finest and most appropriate suppliers to Cloud users, we will continue ranking CSPs in the future based on their key performance indicators.

References

[1] Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems. 2009 Jun 1; 25(6):599-616.

[2] Yeo CS, Buyya R. Service level agreement based allocation of cluster resources: Handling penalty to enhance utility. In2005 IEEE International Conference on Cluster Computing 2005 Sep 27 (pp. 1-10). IEEE.

[3] Quiroz A, Kim H, Parashar M, Gnanasambandam N, Sharma N. Towards autonomic workload provisioning for enterprise grids and clouds. In2009 10th IEEE/ACM International Conference on Grid Computing 2009 Oct 13 (pp. 50-57). IEEE.

[4] Sotomayor B, Keahey K, Foster I. Combining batch execution and leasing using virtual machines. In Proceedings of the 17th international symposium on High performance distributed computing 2008 Jun 23 (pp. 87-96).

[5] Carrera D, Steinder M, Whalley I, Torres J, Ayguadé E. Enabling resource sharing between transactional and batch workloads using dynamic application placement. In Middleware 2008: ACM/IFIP/USENIX 9th International Middleware Conference Leuven, Belgium, December 1-5, 2008 Proceedings 9 2008 (pp. 203-222). Springer Berlin Heidelberg.

[6] Smith M, Schmidt M, Fallenbeck N, Dörnemann T, Schridde C, Freisleben B. Secure on-demand grid computing. Future Generation Computer Systems. 2009 Mar 1; 25(3):315-25.

[7] Kim JK, Siegel HJ, Maciejewski AA, Eigenmann R. Dynamic resource management in energy constrained

heterogeneous computing systems using voltage scaling. IEEE Transactions on Parallel and Distributed Systems. 2008 Jun 27; 19(11):1445-57.

[8] Zhang W, Qian H, Wills CE, Rabinovich M. Agile resource management in a virtualized data center. In Proceedings of the first joint WOSP/SIPEW international conference on Performance engineering 2010 Jan 28 (pp. 129-140).

[9] Wang Z, Zhu X, Padala P, Singhal S. Capacity and performance overhead in dynamic resource allocation to virtual containers. In2007 10th IFIP/IEEE International Symposium on Integrated Network Management 2007 May 21 (pp. 149-158). IEEE.

[10] Aljoumah E, Al-Mousawi F, Ahmad I, Al-Shammri M, Al-Jady Z. SLA in cloud computing architectures: A comprehensive study. Int. J. Grid Distrib. Comput. 2015 Oct 1; 8(5):7-32.

[11] de Asís López-Fuentes F, García-Rodríguez G. Collaborative cloud computing based on P2P networks. In2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA) 2016 Mar 23 (pp. 209-213). IEEE.

[12] Panda SK, Jana PK. SLA-based task scheduling algorithms for heterogeneous multi-cloud environment. The Journal of Supercomputing. 2017 Jun; 73:2730-62.

[13] Hussain A, Aleem M, Iqbal MA, Islam MA. SLA-RALBA: cost-efficient and resource-aware load balancing algorithm for cloud computing. The Journal of Supercomputing. 2019 Oct; 75(10):6777-803.

[14] Wang Y, Tao X, Zhao F, Tian B, Vera Venkata Sai AM. SLA-aware resource scheduling algorithm for cloud storage. EURASIP Journal on Wireless Communications and Networking. 2020 Dec; 2020:1-0.

[15] Saini M, Mangla N. Multi-objective genetic algorithm for job planning in cloud environment. In2021 8th International Conference on Computing for Sustainable Global Development (INDIACom) 2021 Mar 17 (pp. 887-890). IEEE.

[16] Saini M, Mangla N. Performance evaluation of workflow scheduling using meta-heuristic optimization. In AIP Conference Proceedings 2023 Jun 15 (Vol. 2782, No. 1). AIP Publishing.

[17] Mangalampalli S, Swain SK, Karri GR, Mishra S. SLA aware task-scheduling algorithm in cloud computing using whale optimization algorithm. Scientific Programming. 2023 Apr 20; 2023.

[18] Lan S, Duan Z, Lu S, Tan B, Chen S, Liang Y, Chen S. SLA-ORECS: an SLA-oriented framework for reallocating resources in edge-cloud systems. Journal of Cloud Computing. 2024 Jan 15; 13(1):18.