Innovations in Intelligent Systems and Advanced Engineering
Vol. 1, No. 2, 2025 (August)

Efficient Resource Cloud Formation Using Virtualization in SLA-Based Cloud Technologies

R. Meenakshi
Department of Computer Science and Engineering, Chennai Institute of Technology, Chennai, Tamil Nadu, India
meenakshir@citchennai.net

Senthil Kumar Seeni
Department of Mobile Application Development, Cognizant Technology Solutions, Buffalo Grove, Illinois, USA
sseeni@gmail.com

Lalitha Kalaichelvan
Department of Information Technology, Panimalar Engineering College, Chennai, Tamil Nadu, India
lalithapecit@gmail.com

S. Durga Devi
Department of Computer Science and Engineering, Vel Tech High Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Chennai, Tamil Nadu, India
sdurgadevi@gmail.com

B. Beaula Pinky
Department of Artificial Intelligence and Data Science, Sri Shanmugha College of Engineering and Technology, Salem, Tamil Nadu, India
beaulapinky@shanmugha.edu.in

Jayabharathi Ramasamy
Department of Computer Science and Engineering, RMK College of Engineering and Technology, Chennai, India
jayabharathicse@rmkcet.ac.in

K. Navaz
School of Computing, Vel Tech Dr.Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India
navazit@gmail.com

T. Prabakaran
Department of Computer Science and Engineering, Joginpally B.R. Engineering College, Hyderabad, Telangana, India
prabaakar.t@gmail.com

Abstract: Cloud computing has arisen as a viable remedy to the constraints of conventional IT systems, product delivery methodologies, and application management methods, including licensing, configuration, and maintenance. Transitioning from traditional platforms to cloud-based settings diminishes customer-side complexity and expenses, while guaranteeing continuous income for Software as a Service (SaaS) suppliers. Service Level Agreements (SLAs) are instituted as enforceable promises between customers and providers to uphold quality of service (QoS). The main objectives for SaaS providers are to minimize operating expenditures and to maximize Customer Satisfaction Levels (CSL). The present work presents customer-centric SLA algorithms for effective resource allocation, aimed at cost reduction through the minimization of support overheads, fines, and SLA breaches. The suggested system amalgamates user profiles and supplier performance data to encapsulate intricate client needs and tackle heterogeneity across business networks. Customer-specific factors, like enhancement request rates and infrastructure-level metrics such as job initiation timings, are integrated for precise decision-making. Simulation outcomes indicate substantial enhancements, with a 54% reduction in total costs and a 45% drop in SLA breaches, surpassing traditional optimization methods.

Keywords: Cloud computing, SaaS, SLA, CSL, request arrival rate, response time

I. INTRODUCTION

Cloud computing has arisen as a contemporary paradigm for providing computer resources, including processing power, bandwidth, and storage, via a pay-as-you-go basis. It is primarily classified into three service models: SaaS, Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). SaaS delivers software applications to end-users, IaaS supplies automated infrastructure via on-demand provisioning of Virtual Machines (VMs), and PaaS furnishes software development, deployment tools, and execution management services, connecting SaaS and IaaS.

Before the advent of cloud computing, particularly in the early stages of web-based business software, system administration was relatively simple, with efficiency primarily measured in terms of resource utilization time [3]. Over time, however, software complexity has increased, leading to greater logistical challenges. Businesses have increasingly recognized the advantages of outsourcing applications to cloud-enabled third-party SaaS providers due to the following reasons [4]:

- Reduced operating expenses due to increased system complexity and high maintenance complexity.
- Elimination of upfront investment in costly software licenses and hardware, as businesses can instead evaluate market value and pay for services on demand [5].

SaaS providers handle upgrades and new releases while users use continuously maintained apps. Due to its ease, scalability, and cost-effectiveness, the SaaS model is extensively used in banking and financial services [9]. These installations depend on SLAs. An SLA is a formal contract between consumers and suppliers that specifies QoS. Any party that violates SLA provisions shall pay contractual penalties. SaaS companies assign dedicated VMs to each client to meet SLA response time requirements [6]. During low-demand periods, hardware resources may go unused, resulting in inefficiencies.

There is little literature on SLA-based cost optimisation and CSL for SaaS companies. Most study on IaaS providers has concentrated on market-driven models. Many studies have ignored customer-driven management, which allocates resources dynamically depending on user demand. Some algorithms have been presented to lower overall expenses and SLA breaches; however, they generally ignore SaaS provider-essential customer characteristics like firm size [7]. CSL is directly impacted by SLA violations, particularly when response times exceed those specified in the agreement. Since SLA breaches are penalized, providers must carefully manage resources to maintain compliance. The concept of Service Quality Improvement (SQI) has also been explored, which differs from raw response time by evaluating service enhancements. However, SQI is considered a secondary factor, as many consumers may not notice improvements beyond the promised QoS. Therefore, while SQI can influence CSL, it is often excluded from optimization models, with studies instead compiling supporting statistics.

II. LITERATURE SURVEY

Market-driven distribution analysis began in the early 1980s. Several resource distribution approaches were developed based on demand, with a fixed number of resources considered. In the provision of complex services, user-driven SLA-based financial resource allocation plays a significant role. Furthermore, patterns of service usage and demand forecasting are closely related to this work [8]. Notwithstanding the collapse of the e-commerce bubble in the late 1990s, the field of web usage mining (WUM) has persistently advanced. WUM employs data mining methodologies on website clickstream data to extract consumer behaviour insights. Data in this domain are classified into four categories: content, structure, usage, and user profile [9]. The initial three pertain to websites, although are not immediately linked to e-commerce transactions.

Three major types of prediction algorithms are commonly applied to these data categories: historical, sequence-based, and Markov-based methods. Customer profiles and historical approaches are also used for forecasting transaction-based behaviour in companies, often as a means of measuring credit levels rather than relying purely on algorithm design. The examined literature encompasses many pertinent studies in grid and cloud computing, specifically focussing on resource allocation and SLA management. A new strategy for numerical

grid preparation is explained [10]. Their efforts focused on short-term research studies, whereas transaction-based operations were designated for extended periods. Nevertheless, in consumer-driven contexts, results remain uncertain. Assessment methodologies vary, with certain methods concentrating on reaction time and utilisation, whilst others prioritise cost and the frequency of SLA breaches.

Numerous methods have been suggested for market-based resource allocation in grid computing. Two significant tactics are the State Plan, which distributes resources according to the existing system state, and Pre-emptive Tactics, which enable the reassignment of jobs to different resources for enhanced performance [11]. Both studies highlighted market-oriented resource distribution while addressing individual input tasks with quality-of-service time restrictions under fixed resource availability. A QoS-oriented work scheduling technique was devised in grid computing, with throughput identified as a critical QoS criterion. The approach emphasized the early completion of jobs. This aligns with the focus of our study, which also aims at cost control by considering both consumer and supplier QoS parameters.

Another work proposed minimizing resource consumption and handling requests through a predictive scheduling framework that ensures task completion within deadlines [12]. However, their approach targeted short-term, computationally intensive applications, while our work focuses on long-term, data-intensive transactions. Furthermore, our model incorporates penalties and economic objectives that were not addressed in their work [13]. An SLA-based hierarchical scheduling algorithm is discussed for distributed streaming services and evaluated different SLA-based heuristic scheduling methods using two metrics: utilization and service revenue (in terms of CPU nodes). In contrast, our work specifically addresses the scheduling of VMs in cloud computing for business applications, where the VM is the fundamental unit of resource allocation [14].

Since cloud computing relies heavily on virtualization, VM placement plays a critical role in efficient resource management and scheduling. Different VM assignment algorithms with infrastructure provisioning and VM placement are explored. Their work developed complex methods for optimizing homogeneous resources [15]. However, these studies did not account for monetary risks or demand volatility. To address such issues, Bobroff proposed a heuristic VM placement scheme to reduce overall costs for SaaS providers, though it was not based on client-driven scenarios.

III. PROPOSED SYSTEM

Figure 1 illustrates the SaaS model for customer support in a cloud setting. Clients utilize software services from SaaS suppliers via web-based business apps. The provider implements a three-tier cloud architecture consisting of application, platform, and infrastructure levels to meet these needs. The application layer provides secure services to end users through apps like Customer Relationship Management (CRM) and Enterprise Resource Planning (ERP). The platform layer manages the development and deployment of frameworks, applying mapping and scheduling strategies that convert customer QoS needs into infrastructure-level specifications. This layer also integrates customer profiles and Key Performance Indicators (KPIs) in assessing the QoS assured by the supplier. The infrastructure layer provides virtualization functionalities, encompassing virtual machine administration, resource allocation, and the deployment or decommissioning of virtual machines. Resources may be acquired via IaaS providers like Amazon EC2 and S3 or from internal virtualized clusters, with efficiency attained by reducing active VM utilization. The model encapsulates participants, their goals, tasks, and limitations. Essentially, SaaS companies provide applications on a lease basis to save operational expenses and mitigate SLA breaches.

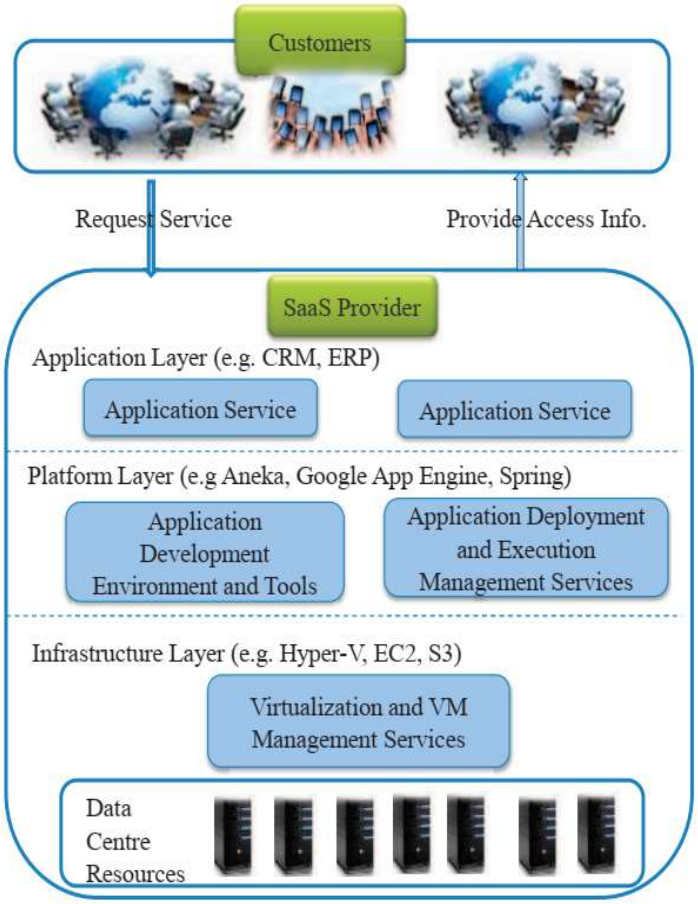

Figure 1: Proposed System Architecture

This is achieved by providing customer-oriented SLA utility equations for web-based business applications. For example, consider a SaaS provider, Company X, that offers three editions of CRM or ERP software services: Standard, Technical, and Enterprise, each with a fixed price. New SaaS providers often follow a similar service model, such as "Compere ERP."

In this model, services are allocated to the client based on demand. For instance, when Company Y submits a request for a "first-time rent" (Standard edition) along with additional accounts, the SaaS provider provisions the requested services. Company Y may later upgrade to a higher version or add supplementary accounts. In such cases, the contents of the existing VM are often migrated, and a new VM is generated. These on-demand customer requirements are handled by the provider in compliance with the SLA.

The SLA defines pre-configured provider settings and QoS parameters guaranteed to the customer, which include:

- **Product Edition (p):** Specifies the type of software package available to customers. For example, SaaS X offers Standard, Technical, and Enterprise editions.

- **Application Form (j):** Sets consumer request type. It might be a new service lease or upgrade. Service updates might add accounts or enhance products. If the client wants to downsize, they must cancel the contract and reorder.

- **Contract Length (cl):** Defines the period during which the customer is authorised to utilise the application.

- **Number of Customers (a):** Denotes the present quantity of client accounts. The maximum quantity is contingent upon the chosen product edition.

- **Number of Records (n):** Indicates the mean quantity of records per customer during a transaction, which may influence the duration of service update processing. This parameter is established in the SLA.

- **Response Time (respT):** Maximum time the supplier can take to process a client's request. SLA violations occur when response time surpasses this limit. Four useServ response time categories have different significance and SLA levels.

- **Penalty Terms:** Every SLA breach costs the SaaS provider dependent on how long it takes to reply to the customer. Different penalty rates are applied depending on the type of order. The corporate rate specifies the financial charges per unit of time for deviations in service delivery.

IV. RESULTS AND DISCUSSION

According to company size, *CompTypeValue* is used to categorize organizations. During identity registration and security verification, businesses are classified as follows: *CompTypeValue = 1* for small enterprises, 2 for medium-sized companies, and 3 for large corporations. This simplified scale is adopted for consistency during evaluation, instead of using values such as 10, 20, or 30. When determining the credit rating, the company type is considered, since larger firms contribute greater value to the SaaS provider's market share.

The credit rating element is determined by historical service update requests and the customer's present conduct. The most recent update is indicated as a Boolean value: "true" if an update request was sent, and "false" if not. The current update is quantified in tangible metrics, such the quantity of accounts or the specific type of service enhancement solicited.

The credit level element is further enhanced by utilising the ratio of current *ValueShift* to prospective *ValueShift*. This method rectifies discrepancies and authenticates client behaviour utilising both historical records and corroborated data. Providers must also account for potential future demand, since many customers indicate higher anticipated requirements during initial planning. This enables providers to prepare resources in advance for "high demand" clients. The td delay is defined as the difference between the SLA-defined response time and the actual response time observed. Delays (i.e., SLA violations) may occur in four cases:

- During "first rent," where long initialization times may cause delays.

- When additional accounts are requested.

- When product editions are upgraded.

- When service usage suffers from system performance degradation beyond the provider's control.

Performance can be evaluated using two approaches:

- Macro-average efficiency, which treats all customers equally regardless of activity levels.

- Micro-average efficiency, which gives greater weight to highly active users who generate more traffic.

Behavioural models based on specific user groups may lack precision compared to global models, since they focus on narrower details. Therefore, performance is often measured as the average per order. Additionally,

penalty costs caused by response time delays must be considered, as they directly affect customer satisfaction and provider revenue, even if service improvements are not visible to all users. Figure 2 shows that the proposed framework responds quickly once a request is acknowledged.

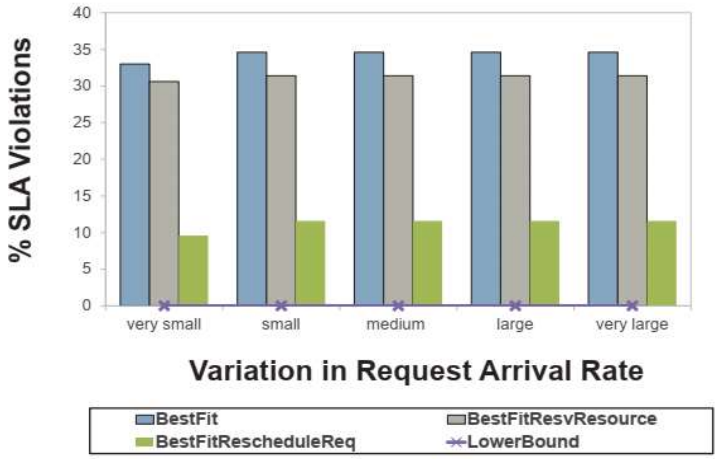

Figure 2: Analysis of Request Arrival Rate

Cloud services enhance efficiency by reducing resource expenditures, contingent upon the quantity and classification of deployed virtual machines (VMs). The suggested approach enhances resource utilisation by reducing the number of active VMs and reallocating work to previously initiated instances wherever feasible to meet increased user demands. To avert SLA breaches, the algorithm guarantees that new requests are not allocated to an already active VM if such allocation jeopardises service quality for current clients. Figure 2 depicts this procedure, with the grey rectangle indicating the inaccessible area and the x-axis representing the IDs of virtual machines (VMs). These VMs are same in kind and align with the same product category as customer *c*. Figure 3 presents the evaluation of the reaction time for the suggested framework.

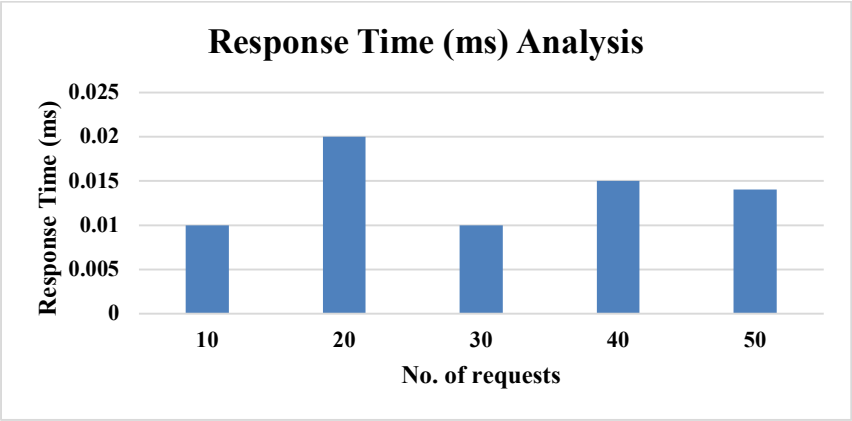

Figure 3: Response Time Analysis of the Proposed Framework

Figure 3 presents the response time analysis of the proposed framework. The response time remains within the range of 0.01 ms to 0.02 ms for all numbers of requests. Even as the number of requests increases, the framework maintains an average response time of approximately 0.01 ms. This demonstrates that the proposed

framework consistently delivers prompt responses under high-demand conditions, thereby enhancing the overall system performance.

IV. CONCLUSION

Clients often want three primary kinds of on-demand services in the public cloud: SaaS, apps as a service, and cloud infrastructure. This study aims to manage resources for SaaS providers to save operating costs while maximizing CSL by decreasing SLA breaches. The research tackled the problems presented in the introduction by analyzing customer profiles and KPI indicators to accomplish this objective. Mapping and scheduling techniques were utilized to address intricate needs and resource diversity. Two customer-centric algorithms were created, integrating several QoS constraints like arrival rate, service initiation time, and penalty rate. Additionally, five resource reservation strategies were presented: one dynamic and four fixed-percentage methods to ascertain the best quantity of services to reserve to enhance performance across QoS metrics. The results indicate that the suggested framework exhibits enhanced performance regarding request arrival rate and response time. The dynamic reservation technique surpassed the fixed-percentage solutions in terms of overall cost, the number of active VMs, and the rate of SLA violations amid changes in service demand.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

REFERENCES

[1]. Y.C. Lee, C. Wang, A.Y. Zomaya, and B.B. Zhou, “Profit-driven Service Request Scheduling in Clouds,” In Proceedings of the 10th International Symposium on Cluster, Cloud and Grid Computing, pp.15-24, 2010.

[2]. O. F. Rana, M. Warnier, T. B. Quillinan, F. Brazier, and D. Cojo carasu, “Managing Violations in Service level agreements,” In Proceedings of the 5th International Workshop on Grid Economics and Business Models, pp. 349-358, 2008.

[3]. D.E. Irwin, and L.E. Grit, and J.S. Chase, “Balancing Risk and Reward in a Market-based Task Service,” In Proceedings of the 13th International Symposium on High Performance Distributed Computing, pp. 160-169, 2004.

[4]. Y. Yemini, “Selfish Optimization in Computer Networks Processing,” In Proceedings of the 20th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes, vol. 66, pp. 46-51, 2014.

[5]. I. Popovici, and J. Wiles, “Profitable Services in an Uncertain World” In Proceeding of the18th Conference on Supercomputing, pp. 36-46, 2005.

[6]. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility, Future Generation Computer Systems,” Future Generation Computer Systems, vol. 25, no. 6, pp. 599-616, 2009.

[7]. D. Parkhill, “The Challenge of the Computer Utility,” Addison-Wesley Educational Publishers Inc., USA, 1966.

[8]. M. A. Vouk, “Cloud Computing-Issues, Research and Implementation”. In Proceedings of 30th International Conference on Information Technology Interfaces, Dubrovnik, Croatia

[9]. T. Gad, "Why Traditional Enterprise Software Sales Fail".http://www.sandhill.com/ opinion/editorial_ print.PHP?id=307. Referenced on March 6, 2010

[10]. Y. Fu, and A. Vahdat, “SLA Based Distributed Resource Allocation for Streaming Hosting Systems”. http://issg.cs.duke.edu. Referenced on 6th Dec 2010

[11]. V. Yarmolenko, and R. Sakellariou, “An Evaluation of Heuristics for SLA Based Parallel Job Scheduling,” In Proceedings of the 3rd High Performance Grid Computing Workshop, pp. 1-8, 2006.

[12]. L. Wu, S. K. Garg, and R. Buyya, “SLA-based Resource Allocation for Software as a Service Provider (SaaS) in Cloud Computing Environments,” In Proceedings of the 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing.

[13]. R. Sakellariou, and V. Yarmolenko, “On the Flexibility of WS-Agreement for Job Submission,” In Workshop on Middleware for Grid Computing, pp. 1-6, 2005.
[14]. V. Yarmolenko, R. Sakellariou, D. Ouelhadj, and J. M. Garibaldi, “SLA Based Job Scheduling: A Case Study on Policies for Negotiation with Resources,” In AHM2005, pp. 20-22, 2005
[15]. Z. Duan, Z. Zhang, and Y. Hou, “Service Overlay Networks: SLAs, QoS, and Bandwidth Provisioning,” In IEEE/ACM Trans. on Networking, vol. 11, no. 6, 2003