A Lightweight Cloud Resource Management Framework

Aaqif Afzaal Abbasi*, Amber Sultan, Syed Baqar Hussain, and Tayyaba Anees
School of Systems and Technology, University of Management and Technology, Lahore, Pakistan
*Correspondence: aaqif.afzaal@yahoo.com, amber.sultan.ahmed@gmail.com, baqar1@gmail.com, tayyaba.anees@umt.edu.pk

Citation | Abbasi, A. A, Sultan. A, Hussain. S. B, Anees. T, "A Lightweight Cloud Resource Management Framework", IJIST, Vol. 8, Issue. 2 pp 698-707, April 2026
Received | March 15, 2026 **Revised |** April 17, 2026 **Accepted |** April 22, 2026 **Published |** April 27, 2026.

Cloud infrastructures provide computing and virtualization services where computing resources like memory, storage, and processing services are provided on demand. The management of these resources is a challenging task, and their execution in public and private cloud environments is often challenging. Existing approaches suggest the use of control techniques to resolve resource allocation problems. However, adoption of these techniques often leads to issues concerning reliance on the network service provider, service level agreements (SLAs) compliance, and data locality, etc. This paper presents a cloud resource administration framework. It aims at enhancing the performance capacity of a system by optimizing admission control and scheduling strategies. We begin by first analyzing the key issues in resource administration. We then present its architecture, followed by its design principles, policies, scenarios, and performance isolation scheme. The framework manages incoming service requests to address resource congestion problems. Its scheduling policy helps in administering load-synchronization issues. We also construct our framework's scheduling policy and employ the Software-defined networking (SDN) concept for performance isolation. Then we rigorously implemented a prototype based on our scenarios and verified our system's effectiveness. Based on our analysis, we identify that the proposed framework demonstrates significant improvement in workload admission, resource balancing, and congestion management up to 30% with slight performance degradation under controlled parameters.

Keywords: Resource allocation, Admission control, Scheduling, Cloud computing, Virtualization, Resource management, SDN.

Introduction:

Cloud computing is considered a highly useful computing paradigm [1]. Large-scale computation and I/O resource management tasks in both public and private clouds. The ability of cloud management software to effectively use the underlying hardware resources while maintaining the required performance isolation is crucial to its success [2]. Consumers anticipate quality of service (QoS) controls for tenant virtual machines (VMs) from cloud service providers. Because of this, cloud-scale resource management necessitates that the management platform offer a wide range of resource controls that strike a compromise between tenants' quality of service and datacenters' overall resource efficiency.

The adoption of virtualized data centers is increasing. However, real-time limitations and synchronization issues prevent current hypervisors from offering sufficient support for them, leading to frequent deadline misses and significant performance loss. The effectiveness of virtualization technologies used in cloud environments for resource consolidation and management services is one of its key features. By consolidating several independent physical machines into a virtualized environment through the use of hypervisors in a cluster environment, fewer physical resources are needed than ever before [3]. Although this makes things better, it is frequently insufficient. This is because hiding services within clouds enables us to consider how users are supposed to utilize them. Customers are kept in the dark regarding the specifics of service delivery. In essence, consumers concentrate on what matters most to them while using a service. In a similar vein, cloud services only concentrate on facets of their industry without providing transparency to subscribers. This division of responsibilities offers a very sophisticated and practical method for creating services and communicating with them. In essence, it establishes the interface between service suppliers and service users.

Objectives:

We present a cloud resource management framework that improves resource allocation and provision strategy in virtualized environments. It consists of 3 major components: Admission controller, Scheduler, and Performance isolator. The first component (admission controller) assigns services and applications to hosts only if the hosts qualify for having sufficient processing capacity. This assists in reducing VM gaps [4]. The scheduler implements fairness and load-balancing features in ascending order. Finally, a careful isolation of network (functions) by employing SDN concepts using VLAN IDs [5] is performed. This stringent combinatorial exercise assists in improving the overall facilitation of the resource provisioning mechanism in virtualized environments. In the following section give an outlook of our paper's contribution.

Contributions:

We present a cloud resource management framework and explicate its features. It administers admission control and scheduling functions by employing admission control and Scheduling policies. Results demonstrate that the framework is more efficient in terms of memory usage and CPU utilization. Furthermore, the employment of SDNs in performance isolation (Section IV) makes the framework suitable (especially) for memory-intensive cloud I/O requests.

Organization:

The manuscript is organized as follows. In Section II, related developments are discussed. A brief outline of the system design is given in Section III. In Section IV, we implement and evaluate the proposed framework. A comprehensive evaluation of the framework is presented. Finally, Section V concludes the paper.

Related Work:

Organizations with large-scale cloud deployments might need more powerful cloud management tools with certain features, like the capacity to oversee procedures and functions.

With features like self-monitoring [6], an explicit alerting system [7], failover, and self-healing, high-end cloud management systems can automatically manage system faults [8].

For complex applications, cloud management systems also include frameworks for workflow mapping. In this section, we describe the important developments made in cloud resource management systems. We also review software-defined cloud solutions to resource management challenges in virtualized environments.

The performance of a gossip-based protocol [9] that computes a heuristic solution to the resource allocation problem was assessed using simulations. The protocol accomplishes objectives, such as equitable distribution of resources among sites, effective adaptability to load fluctuations, and middleware layer scalability with respect to the number of cloud machines and hosted sites.

Cloud datacenters must meet the demand for computing services from competing apps to handle SLA-related resource allocation mechanisms [10]. Specifically, user applications are getting increasingly complicated and require more than one service to run. Virtualization concerns are the most commonly occurring problems that customers encounter, partly because they are unable to see the cloud and must thus seek assistance from cloud operators. According to the surveys, clouds should create solutions to automate operator tasks in order to provide more efficient assistance.

In [11], the resource provisioning strategies for lowering network overheads are discussed. Infrastructure as a Service (IaaS), multi-level scheduling, and advanced bookings are some of these strategies. The study examines the benefits and drawbacks of these methods in terms of usability, performance, and cost. In [12][13], the authors discuss Nephele. It is a data processing framework. The ability to automatically allocate and de-allocate virtual machines during process execution is made possible by the performance evaluation of the suggested framework. This lowers processing costs and enhances overall resource use. In [14], an on-demand grid computing architecture design is put out. In [15], a profile-based method for capturing expert knowledge of scaling applications was put out, allowing for the more effective provision of additional resources that are needed. In [16], an optimization framework is presented. The framework took into account several client QoS classes when workloads were unpredictable (e.g., needs of computing resources). Online forecasting algorithms are used to estimate the workload arrival trend. One of an infrastructure provider's most important requirements is decision-making. Notably, it impacts the elasticity criteria on future capacity requirements that are unknown at the time-of-service initiation. Admitting new services can occasionally make it more likely that those that are currently in place will fail. In a similar vein, a stringent acceptance process for incoming services could lead to more rejections. In order to respond to unforeseen load changes more quickly than the rate at which the load is changing, elasticity choices should ideally be able to predict changes in a service's load in advance [17]. This calls for quick and dependable use and prediction techniques. In virtualized environments, admission control services determine whether a group of services is permitted to enter a virtualized infrastructure. Numerous aspects, including services, infrastructure, QoS, policies, and expenses, are the foundation of these schemes.

System Design:

Any cloud management software's capacity to use the underlying hardware resources with flexibility and scalability is crucial to its effectiveness. With the introduction of software-defined clouds (SDCs) in particular and SDN in general, network administrators can easily enforce system-level regulations to resolve the long-standing variety of system constraints. Virtualized systems manage and enable resource administration capabilities to efficiently use the underlying network resources by

Efficient resource utilization

Supporting incoming requests.

Resource allocation per requirements.
Variable QoS assignment to VMs

Due to increasing complexity, network administration gets complicated. We highlight a few of the many challenges in virtualization environments below.

Resource pre-emption: Service providers want to take on as many services as they can in order to maximize profitability. To deliver these services using pre-emption-aware schemes, effective methods are needed.

Oversubscription: Oversubscription can take advantage of unused cloud resources, but it can also result in overload. In this sense, admission control is crucial. It prevents the effects of oversubscription.

Overhead and overbooking: Tenant VM performance is impacted by VM contention. Effective resource consolidation schemes, which are specifically described in [18], use a variety of overhead reduction techniques to overcome these problems.

The framework employs service admission policies for capacity planning, whilst a scheduling technique policy is developed to focus on avoiding performance degradation due to (short-term) workload fluctuations. We also employ SDN concepts of performance isolation in an attempt to carefully isolate data traffic functions and to ensure a high level of integrity and service quality for services. We hope that the performed work will motivate future research in designing and developing more practical solutions for these core resource management issues. We present a novel three-tier mechanism for admission control and scheduling of service requests. We also implemented SDN management concepts. Our reason for employing SDN is its neatness and simplicity. In summary, our work is unique in the following aspects: We propose an approach for maximizing the overall utility of traffic sources while honoring each source's minimum rate requirements and each link's capacity constraints. We use fairness and prioritization for scheduling real and batch workloads. In addition, we implemented SDN concepts to isolate virtual networks to ensure a high level of integrity and service quality for service requests. The isolation can also improve the overall admitted service request capacity in the system. In view of the above, the proposed 3-fold joint resource allocation strategy can effectively answer resource administration and allocation concerns in virtualization environments.

The challenges for resource-optimized cloud data centers in virtualized environments are immense. In addition to the challenges mentioned in Section II, virtualized environments have further constraints. The proposed framework reduces dependence on conventional admission control and scheduling approaches. In this section, we explain the key terms through which our system can extend its support to existing cloud infrastructures.

Capacity Awareness: In order to reduce overloading, we use SDN-based controls. This assists in defining and extending process capacity and reducing service degradation.

Service and process integrity: Virtualization-based systems share resources. However, they remain separate in the host hypervisor. This feature extends VM functions. We extend this feature further by enabling process handling to be managed at the site level. This helps in achieving performance isolation of services in a clearer way.

Prioritization of services: Setting service priorities reduces or eliminates the need for a thorough and practiced plan. This classification complies with the system's optimization needs and does not break any QoS regulations.

As an important technical requirement for Network Admission Control systems in Network virtualization, an isolation mechanism for a device is required from the internal enterprise (until the user access policy is applied). We therefore employed a VLAN-ID-based performance isolation scheme (by using the SDN concept of isolation) and assigned unique VLAN IDs to individual services.

Our design encourages a global view of workloads by unifying administration for these workloads and thereby providing a clearer picture of the admission control process. In Figure 1, we present the functional layout of the proposed framework. We manage traffic load constraints by improving interactive analytics for admission control requests. This helps in devising a joint optimal solution for admission control. We believe that the QoS of incoming requests must be ensured to improve system performance. The admission controller assigns applications to hosts with need resources.

The proposed framework policy (for admission control requests) involves locating resources that have similar objectives. We address the scheduling of requests by considering a scenario. The requests accepted by the Virtual CPU (VCPU) of the Virtual Machine Manager (VMM) synchronize these requests before processing, as shown in Figure 2. The duration for fulfillment/solving of tasks is dependent on the value of occupied space (level of fullness) of the random-access memory of the hosting machines.

Our proposed scheduler prioritizes and accepts those application requests first, which have maximum load-share capacity. These application requests are granted priority and admitted first. After assigning priorities, we arrange queries in non-decreasing order, where every incoming request (best fitting the resource requirement) is treated first.

Performance isolation methods are employed with the objective that carefully isolated (functions of) networks will not affect other networks' performance. There are different degrees of isolation for isolating systems in virtualized environments.

We isolate physical networks' similar functioning VLANs (See Figure 3). By employing the SDN concept of isolation, we assign unique VLAN IDs to individual services. A VLAN ID consists of a service and a VLAN identification number. This results in developing VLAN-ID-based performance isolation.

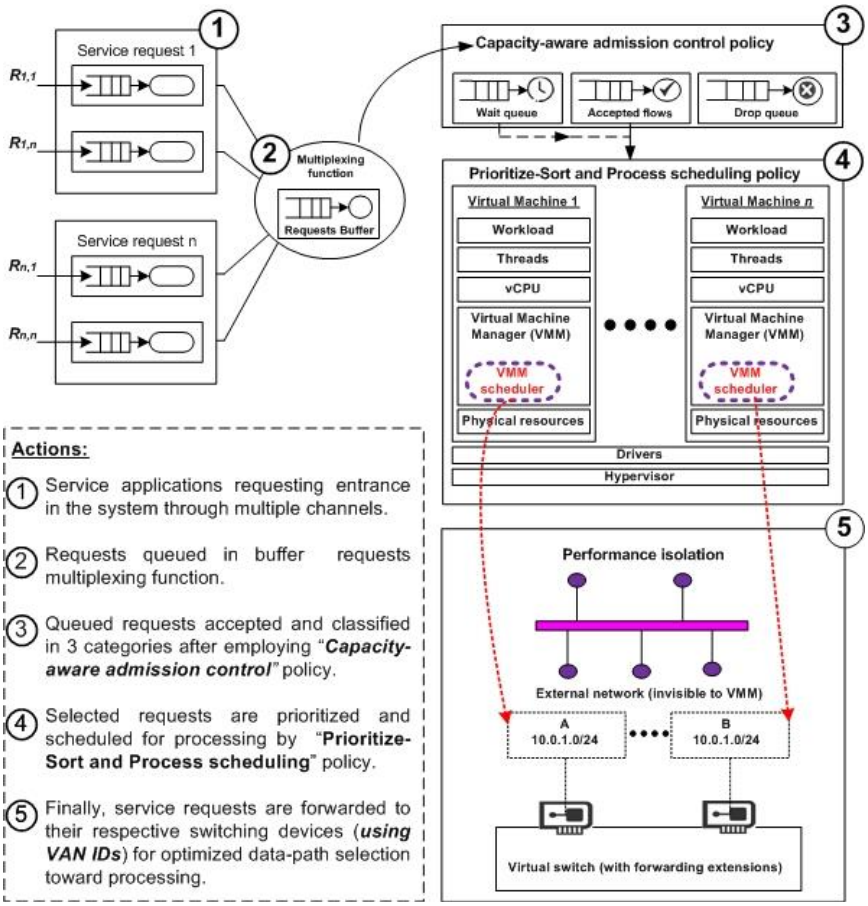


Figure 1. Functional Layout – An Overview

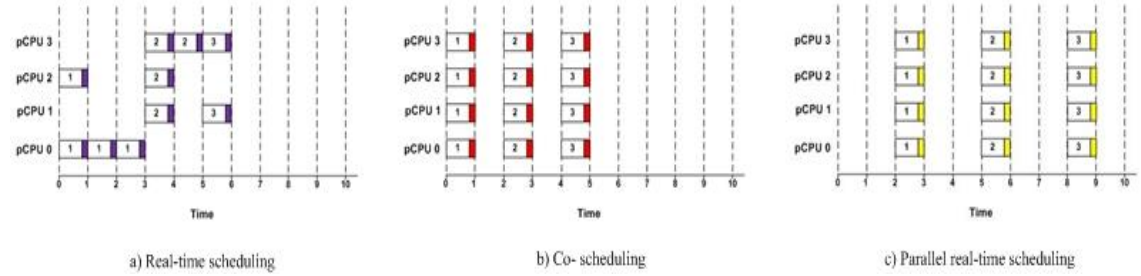


Figure 2. Single VM-Multiple request-based scheduling scenarios

OpenFlow-hybrid switches support both OpenFlow and normal Ethernet switching functions [19] (like VLAN isolation, L2 Ethernet switching, L3 routing, etc.). They use VLAN IDs to determine whether packets are processed via using one path or the other. We implemented it to set up a priority field in VLAN tags. We then use the OFPAT_SET_VLAN_PCP action to update the (3-bit) link priority field in the VLAN tag. In our case, we performed isolation by using 3 VLAN IDs. This allows a .0Our motive for performance isolation is to ensure a high level of integrity and service quality for service requests.

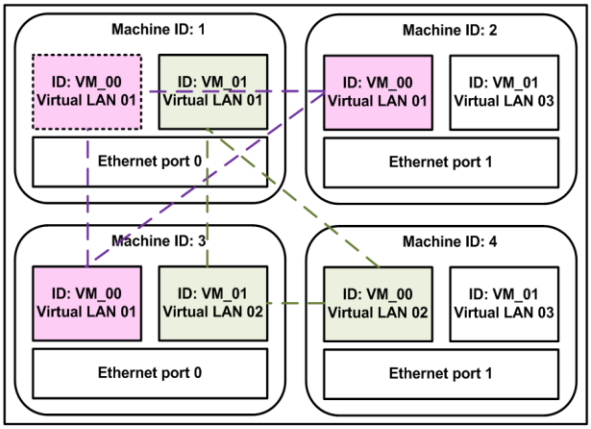


Figure 3. Function Isolation

Implementation and Evaluation:

In this section, we briefly explain our proposed system's implementation, evaluation, and results. We begin by describing the workloads and benchmarks, followed by details regarding experimental setup and results. We also present a short discussion on the outcome of the results.

We conduct several experiments to compare our proposed framework's performance with realistic and capacity-aware resource allocation schemes. Our main purpose behind performing the evaluations is to answer the following questions.

To find the workload execution efficiency (with and without performance isolation).

Performance checks on using 3 benchmarks.

Measure CPU and Memory utilization characteristics.

Check the overheads incurred by the framework.

We execute the performance evaluation by submitting admission control requests and subsequently evaluating the system's CPU and Memory consumption. We employed batch workload in our trials.

A job command file, which usually outlines the important details of the required computational resources and execution environment, is what defines a batch task. Additionally, we measured the execution time of the workload (Figure 4).

Next, we contrast the effectiveness of the suggested framework with capacity-aware and realistic admission control approaches. Product logic is used in the realistic method to

represent demands. In [20][21], a thorough analysis is provided. The capacity-aware admission control system uses real-time values for memory mapping of incoming admission control requests. Our aim in performing an evaluation using 2 parallel ways was to ensure that implementing the isolation scheme would not generate excessive traffic, which may affect the (necessary) operations of services. Table 1 provides a short description of the benchmarks used in our experiments, whereas Table 2 shows a cost comparison (with and without isolation) for employing the isolation scheme on batch workload.

Our benchmarks aim to evaluate admission control behavior for time-varying workloads. These benchmarks are Component Level Threshold Performance (CLTP) [22], SPECweb [23], and HCprobe [24]. We use these benchmarks to measure the task responsiveness of Batch workloads. Moreover, we used these benchmark schemes as a reference for measuring our admission control scheme. It is because the service requests entering tend to relinquish their pCPUs to the VM running on prescribed benchmarks (Figure 5).

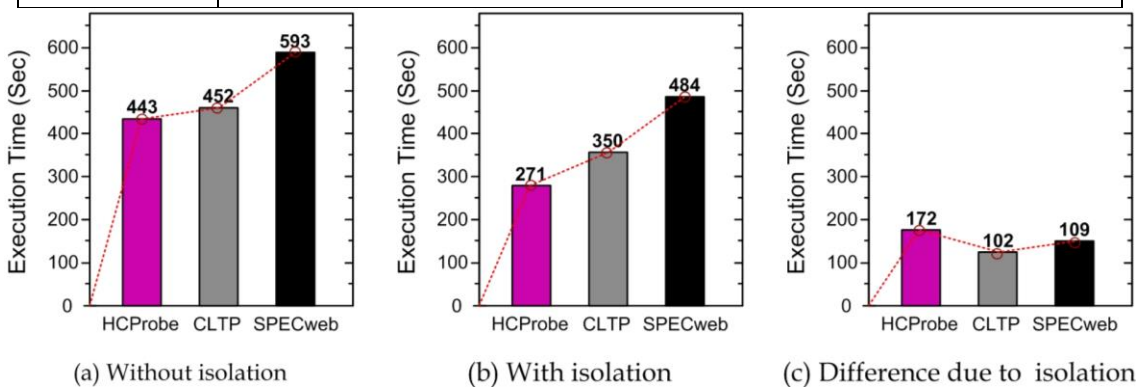
We test the prototype of our suggested system on three Intel Xeon CPUs with eight cores (16 threads). Python 2.7.9 is used in its development. We use 2 x 8-slot SDN-enabled switches, using the Ubuntu 25.10 environment.

We found that network resources are being constrained by realistic and capacity-aware strategies [25][26][27]. These methods improved one capacity's performance (CPU utilization) at the expense of another's (memory usage). As a result, behavior becomes more asymmetrical. The proposed framework, on the other hand, has more symmetrical behavior in terms of simultaneous CPU and memory usage. Its effects on CPU and memory resource usage are seen in Figure 6. Similar performance variations were performed in [28][29][30]. Compared to other strategies, the interference with Xen applications is slight and acceptable (See Figure 6 for overhead comparison with variable scheduling intervals).

The framework can manage a growing volume of application requests. This is because the allocation system differs from utilization. It can be challenging at times to keep resource allocation and use in balance. As a result, giving a busy job a little or large portion of CPU share may not fix the issue, but may cause performance to drop.

Table 1. Benchmarks

Benchmarks	Description
CLTP	Represents characteristics of the (in-use) hardware architecture states.
SPECweb	It calculates the server workload capacity from a connection-based workload to a server workload.
Hcprobe	Written in Haskell, Hcprobe specifies patterns for generating OpenFlow messages for traffic-based benchmarks, profiles, and scenarios.



(a) Without isolation (b) With isolation (c) Difference due to isolation

Figure 4. Workload execution time comparison for sequential read applications

Table 2. Execution costs of the workload

Case	Batch workload	Batch workload with isolation
Accepted services	16	16
Bandwidth	15 Mb/s	13.3 Mb/s
Access time	33.5 sec	32.6 sec

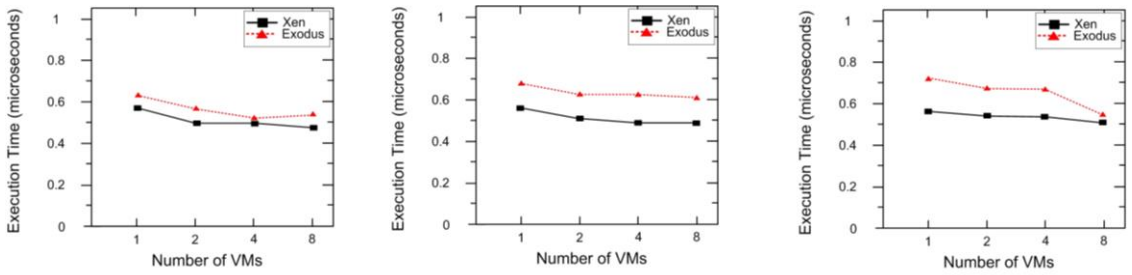
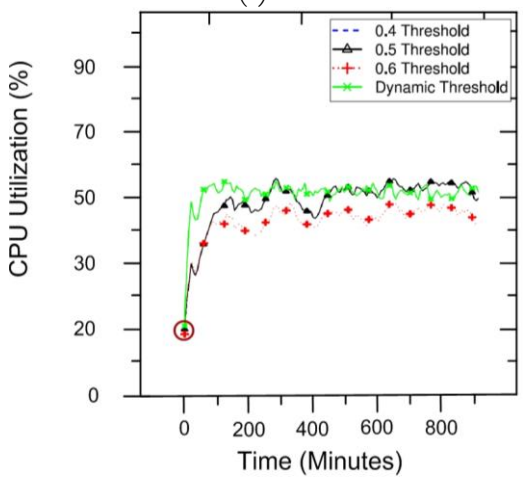


Figure 5. Overhead comparison to Xen, with variable scheduling intervals (a) 5ms, (b) 15 ms, and (c) 30 ms



(a) CPU utilization

Figure 6. System resources comparison

Conclusion:

Cloud resource management systems often perform poorly under overload. This work proposes a resource provisioning strategy for virtualized environments that cloud datacenters face due to the elastic nature of cloud services. Our proposed framework not only improves CPU and memory utilization but also helps in ensuring fairness amongst cloud tenants when they are using the same physical resources.

The cloud management framework employs policies to improve workload distribution and fairness guarantees for workloads and incoming service requests. The policy can be implemented under controlled environments in real data centers. In order to ensure a high level of integrity and QoS, we then implemented a performance isolation scheme using SDNs by assigning unique VLAN IDs to individual service requests. Our proposed framework can be used to handle concurrent applications in small-scale deployments with restricted workload settings. Finally, we use 3 important benchmarks to evaluate our system performance for improved CPU and Memory utilization. Our results clearly indicate that the proposed framework outperforms realistic and capacity-aware resource allocation schemes in terms of CPU and memory utilization. In the future, we plan to use VM-based containerization for consumer-centric applications when hosting sensitive applications.

References:

[1] Aaqif Afzaal Abbasi, Mohammed A.A. Al-Qaness, "Bouncer: A Resource-Aware Admission Control Scheme for Cloud Services," *Electronics*, vol. 8, no. 9, p. 928, 2019, doi: https://doi.org/10.3390/electronics8090928.

[2] Călin Iorgulescu, Reza Azimi, "PerfIso: performance isolation for commercial latency-sensitive services," *USENIX ATC '18 Proc. 2018 USENIX Conf. Usenix Annu. Tech. Conf.*, 2018, [Online]. Available: https://dl.acm.org/doi/10.5555/3277355.3277406

[3] "(PDF) Seawall: Performance isolation for cloud datacenter networks." Accessed: Apr. 30, 2026. [Online]. Available: https://www.researchgate.net/publication/234793767_Seawall_Performance_isolation_for_cloud_datacenter_networks

[4] A. A. Abbasi, S. Sultana, M. A. A. Al-Qaness, A. Hawbani, S. Javed, and S. Kim, "Lightweight virtual machine mapping for data centers," *Proc. - 2019 IEEE Int. Conf. Ubiquitous Comput. Commun. Data Sci. Comput. Intell. Smart Comput. Netw. Serv. IUCC/DSCI/SmartCNS 2019*, pp. 318–322, Oct. 2019, doi: 10.1109/IUCC/DSCI/SMARTCNS.2019.00080.

[5] X. Wang, H. Zhao, M. Guan, C. Guo, and J. Wang, "Research and Implementation of VLAN Based on Service," *GLOBECOM - IEEE Glob. Telecommun. Conf.*, vol. 5, pp. 2932–2936, 2003, doi: 10.1109/GLOCOM.2003.1258771.

[6] Nektarios Deligiannakis, Vassilis Papataxiarhis, "DACCA: Distributed Adaptive Cloud Continuum Architecture," *Futur. Internet*, vol. 18, no. 2, p. 74, 2026, doi: https://doi.org/10.3390/fi18020074.

[7] P. Panindre, S. Acharya, N. Kalidindi and S. Kumar, "Artificial-Intelligence-Integrated Autonomous IoT Alert System for Real-Time Remote Fire and Smoke Detection in Live Video Streams," *IEEE Internet Things J.*, vol. 21, pp. 45133–45149, 2025, doi: 10.1109/JIOT.2025.3598979.

[8] Cheng Ji, Huaiying Luo, "Cloud-Based AI Systems: Leveraging Large Language Models for Intelligent Fault Detection and Autonomous Self-Healing," *arXiv:2505.11743*, 2025, [Online]. Available: https://arxiv.org/abs/2505.11743

[9] A. J. Ganesh, A. M. Kermarrec, and L. Massoulié, "Peer-to-peer membership management for gossip-based protocols," *IEEE Trans. Comput.*, vol. 52, no. 2, pp. 139–149, Feb. 2003, doi: 10.1109/TC.2003.1176982.

[10] J. Guitart, M. Macías, O. Rana, P. Wieder, R. Yahyapour, and W. Ziegler, "SLA-Based Resource Management and Allocation," *Mark. Grid Util. Comput.*, pp. 261–284, Nov. 2009, doi: 10.1002/9780470455432.CH12.

[11] S. Singh, S. Mathur, and S. Kumar Khatri, "Analysis of Scaling in Cloud Infrastructure," *Proc. Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2018*, pp. 1375–1379, Dec. 2018, doi: 10.1109/ICIRCA.2018.8597194.

[12] Nick Weber, David Liou, "Nephele: a cloud platform for simplified, standardized and reproducible microbiome data analysis," *Bioinformatics*, vol. 34, no. 8, pp. 1411–1413, 2018, [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/29028892/

[13] M. Smith, M. Schmidt, "Secure on-demand grid computing," *Futur. Gener. Comput. Syst.*, vol. 25, no. 3, pp. 315–325, 2009, doi: https://doi.org/10.1016/j.future.2008.03.002.

[14] L. M. Vaquero, L. Rodero-Merino, and R. Buyya, "Dynamically scaling applications in the cloud," *Comput. Commun. Rev.*, vol. 41, no. 1, pp. 45–52, Jan. 2011, doi: 10.1145/1925861.1925869;WGROUP:STRING:ACM.

[15] R. Chard *et al.*, "Cost-Aware Cloud Profiling, Prediction, and Provisioning as a Service," *IEEE Cloud Comput.*, vol. 4, no. 4, pp. 48–59, Jul. 2017, doi: 10.1109/MCC.2017.3791025.

[16] M. B. Chhetri, A. R. M. Forkan, Q. B. Vo, S. Nepal, and R. Kowalczyk, "Towards Proactive Risk-Aware Cloud Cost Optimization Leveraging Transient Resources," *IEEE Trans. Serv. Comput.*, vol. 16, no. 4, pp. 3014–3026, Jul. 2023, doi: 10.1109/TSC.2023.3253473.

[17] V. Priya, C. Sathiya Kumar, "Resource scheduling algorithm with load balancing for cloud service provisioning," *Appl. Soft Comput.*, vol. 76, pp. 416–424, 2019, doi: https://doi.org/10.1016/j.asoc.2018.12.021.

[18] Jesus Omana Iglesias, Milan De Cauwer, "Increasing task consolidation efficiency by using more accurate resource estimations," *Futur. Gener. Comput. Syst.*, vol. 56, pp. 407–420, 2016, doi: https://doi.org/10.1016/j.future.2015.08.018.

[19] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006, doi: 10.1109/JSAC.2006.879350.

[20] D. L. Eager, E. D. Lazowska, and J. Zahorjan, "Adaptive Load Sharing in Homogeneous Distributed Systems," *IEEE Trans. Softw. Eng.*, vol. SE-12, no. 5, pp. 662–675, 1986, doi: 10.1109/TSE.1986.6312961.

[21] Ming Tao, Kaoru Ota, "Capacity-aware security access authentication in federated-IoT-enabled V2G networks," *J. Parallel Distrib. Comput.*, vol. 118, no. 1, pp. 107–117, 2018, doi: https://doi.org/10.1016/j.jpdc.2017.09.004.

[22] Y. Liu, A. Fekete, and I. Gorton, "Design-level performance prediction of component-based applications," *IEEE Trans. Softw. Eng.*, vol. 31, no. 11, pp. 928–941, Nov. 2005, doi: 10.1109/TSE.2005.127.

[23] Y. Chen, D. Xie, H. Cheng, X. Yang, H. Nie, and T. Liu, "Evaluating SPECWeb 2009 on High-Density and Low-Energy Sugon Cloud Servers," *Proc. - 2016 3rd Int. Conf. Inf. Sci. Control Eng. ICISCE 2016*, pp. 368–373, Oct. 2016, doi: 10.1109/ICISCE.2016.88.

[24] A. Shalimov, D. Zuikov, D. Zimarina, V. Pashkov, and R. Smeliansky, "Advanced study of SDN/OpenFlow controllers," *ACM Int. Conf. Proceeding Ser.*, 2013, doi: 10.1145/2556610.2556621;PAGE:STRING:ARTICLE/CHAPTER.

[25] A. A. Abbasi and M. A. A. Al-qaness, "Services Management in the Digital Era—The Cloud Computing Perspective," pp. 97–105, 2023, doi: 10.1007/978-3-031-28106-8_7.

[26] Rafael Weingärtner, Gabriel Beims Bräscher, "Cloud resource management: A survey on forecasting and profiling models," *J. Netw. Comput. Appl.*, vol. 47, pp. 99–106, 2015, doi: https://doi.org/10.1016/j.jnca.2014.09.018.

[27] "Multi-cloud resource management: cloud service interfacing | Journal of Cloud Computing | Springer Nature Link." Accessed: May 12, 2026. [Online]. Available: https://link.springer.com/article/10.1186/2192-113X-3-3

[28] A. J. Younge, G. Von Laszewski, L. Wang, S. Lopez-Alarcon, and W. Carithers, "Efficient resource management for cloud computing environments," *2010 Int. Conf. Green Comput. Green Comp 2010*, pp. 357–364, 2010, doi: 10.1109/GREENCOMP.2010.5598294.

[29] Joseph Nathanael Witanto, Hyotaek Lim, "Adaptive selection of dynamic VM consolidation algorithm using neural network for cloud resource management," *Futur. Gener. Comput. Syst.*, vol. 87, pp. 35–42, 2018, doi: https://doi.org/10.1016/j.future.2018.04.075.

[30] M. A. Khan, A. Paplinski, A. M. Khan, M. Murshed, and R. Buyya, "Dynamic virtual machine consolidation algorithms for energy-efficient cloud resource management: A review," *Sustain. Cloud Energy Serv. Princ. Pract.*, pp. 135–165, Sep. 2017, doi: 10.1007/978-3-319-62238-5_6/SAVE-RESEARCH.

Copyright © by authors and 50Sea. This work is licensed under the Creative Commons Attribution 4.0 International License.