

Quality of Service Challenges and Solutions in Multi-Tenant Community Clouds: A Comprehensive Survey

Anand Kumar H ¹, Sunil Kumar G ², Mustafa Basthikodi ¹

¹Department of Computer Science & Engineering, Sahyadri College of Engineering & Management, Mangaluru, VTU, Belagavi,

² Department of Computer Science & Engineering, University of Visvesvaraya College of Engineering, Bengaluru.

Abstract - Community cloud computing has emerged as an effective deployment model for organizations that share common security policies, governance frameworks, and regulatory requirements while benefiting from collaborative resource sharing. By enabling multiple organizations to utilize a common cloud infrastructure, community clouds improve resource utilization, reduce operational costs, and facilitate secure collaboration. The integration of multi-tenancy further enhances infrastructure efficiency by allowing multiple tenants to share computing resources through logical isolation mechanisms implemented using virtualization and containerization technologies. However, the shared nature of these environments introduces significant challenges in maintaining consistent Quality of Service (QoS), including resource contention, workload interference, heterogeneous application demands, noisy-neighbor effects, and stringent Service Level Agreement (SLA) requirements. Consequently, effective QoS management has become a critical requirement for ensuring reliable, scalable, and fair service delivery in multi-tenant community cloud environments. This survey presents a comprehensive review of QoS management techniques developed for multi-tenant community clouds. It examines the evolution of community cloud computing, multi-tenancy models, virtualization technologies, and key QoS attributes such as response time, throughput, availability, reliability, scalability, latency, energy efficiency, fairness, and SLA compliance. Furthermore, the survey critically reviews recent advances in resource scheduling, dynamic resource allocation, load balancing, container orchestration, Software-Defined Networking (SDN), and Artificial Intelligence (AI)-based cloud optimization. Existing approaches are comparatively analyzed to identify their strengths, limitations, and applicability across different cloud scenarios. Finally, emerging research directions, including autonomous cloud management, edge-cloud integration, explainable artificial intelligence, and sustainability-aware resource management, are discussed to highlight future opportunities for developing adaptive, intelligent, and scalable QoS frameworks for next-generation community cloud infrastructures.

Keywords - Community Cloud Computing, Multi-Tenancy, Quality of Service (QoS), Cloud Computing, Resource Allocation, Service Level Agreement (SLA), Virtualization, Containerization, Kubernetes, Software-Defined Networking (SDN), Artificial Intelligence (AI).

I. INTRODUCTION

Cloud computing has fundamentally transformed the delivery of computing resources by enabling on-demand access to configurable computing infrastructure, storage, networking, and software services through Internet-based platforms. Instead of maintaining dedicated on-premises infrastructure, organizations can acquire computational resources according to workload requirements using a pay-per-use model, thereby reducing capital investment while improving scalability, elasticity, and operational flexibility. The cloud paradigm has become a key technological enabler for digital transformation across healthcare, education, banking, government, manufacturing, scientific research, and enterprise information systems because of its ability to provide ubiquitous network access, rapid elasticity, resource pooling, and measured services [1].

The National Institute of Standards and Technology (NIST) defines cloud computing as a model that enables ubiquitous, convenient, and on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. Based on ownership and accessibility, cloud infrastructures are categorized into four deployment models: public, private, hybrid, and community clouds. Each deployment model addresses different organizational requirements concerning security, governance, operational control, and infrastructure ownership. Public clouds emphasize scalability and cost efficiency, whereas

private clouds provide enhanced security and administrative control. Hybrid clouds integrate both deployment models to achieve operational flexibility. Community clouds, however, represent a specialized deployment model designed for organizations that share common operational objectives, regulatory requirements, security policies, or compliance standards [1], [2].

The increasing adoption of community cloud computing is primarily driven by the growing demand for collaborative computing environments that support secure resource sharing among organizations operating under similar governance frameworks. Institutions in healthcare, higher education, government administration, financial services, and scientific research frequently require collaborative access to computing infrastructure while maintaining strict regulatory compliance and data confidentiality. Deploying separate private cloud infrastructures for every participating organization often results in underutilized resources, increased operational expenditure, and greater administrative complexity. Community cloud infrastructures address these limitations by allowing participating organizations to share computing resources while preserving logical isolation of applications and data, thereby improving infrastructure utilization and reducing operational costs [2], [3].

A fundamental architectural characteristic enabling efficient community cloud operation is **multi-tenancy**, which allows multiple independent organizations, commonly referred to as tenants, to share the same physical computing infrastructure

while maintaining logical separation of applications, services, and datasets. Multi-tenancy improves infrastructure utilization, simplifies maintenance, reduces operational expenditure, and enhances scalability through efficient resource sharing. Modern cloud platforms implement multi-tenancy using virtualization, virtual machines, containerization, and software-defined infrastructure technologies that dynamically allocate computational resources according to workload demand while maintaining tenant isolation [3], [4].

Although multi-tenancy significantly improves resource utilization and cost efficiency, it simultaneously introduces complex resource management challenges that directly affect system performance. Since multiple tenants compete for shared computational resources such as processors, memory, storage systems, and network bandwidth, cloud environments frequently experience workload interference, resource contention, and performance variability. One of the most widely recognized challenges is the **noisy-neighbor effect**, in which excessive resource consumption by one tenant negatively influences the performance experienced by neighboring tenants executing on the same physical infrastructure. Such interference may increase response time, reduce throughput, degrade service availability, and ultimately violate contractual Service Level Agreements established between cloud providers and tenants [5].

Maintaining **Quality of Service (QoS)** therefore represents one of the most critical challenges in multi-tenant community cloud environments. QoS describes the capability of a cloud platform to consistently satisfy predefined performance objectives while efficiently utilizing available computational resources. In cloud computing, QoS is commonly evaluated using several performance metrics, including response time, throughput, latency, availability, reliability, scalability, elasticity, resource utilization, fault tolerance, energy efficiency, fairness, and SLA compliance [6], [7]. Achieving acceptable QoS requires intelligent resource management frameworks capable of dynamically adapting to continuously changing workload conditions while ensuring equitable resource allocation among multiple tenants. Consequently, cloud providers increasingly employ adaptive scheduling algorithms, dynamic resource provisioning, intelligent load balancing, workload prediction, and automated monitoring mechanisms to optimize cloud performance without compromising service reliability.

Recent technological developments have significantly expanded the scope of QoS management by integrating cloud-native technologies with intelligent resource optimization techniques. Software-Defined Networking (SDN), Network Function Virtualization (NFV), Kubernetes-based container orchestration, edge computing, and Artificial Intelligence (AI) have enabled cloud infrastructures to make adaptive and data-driven resource allocation decisions. Machine learning algorithms are increasingly employed to predict workload patterns, forecast resource demand, detect anomalies, optimize scheduling policies, and automate cloud resource management. These intelligent approaches improve infrastructure utilization while reducing SLA violations, minimizing operational costs, and enhancing overall service reliability [8]–[10].

Despite extensive research on cloud resource management, virtualization, scheduling algorithms, and SLA-aware provisioning, existing studies remain fragmented across different deployment models and application domains. Most published surveys primarily concentrate on public cloud infrastructures or generalized cloud resource management, providing limited discussion of QoS challenges specific to multi-tenant community cloud environments. Furthermore, emerging research areas—including fairness-aware scheduling, explainable artificial intelligence, autonomous cloud orchestration, sustainability-aware resource management, and edge-cloud integration—have received comparatively limited attention within current survey literature. Therefore, a comprehensive survey that consolidates these developments is required to provide researchers and practitioners with a unified understanding of existing QoS management strategies while identifying future research opportunities for adaptive, intelligent, and sustainable community cloud ecosystems.

The remainder of this paper is organized as follows. Section II presents the architectural characteristics of community cloud computing and multi-tenancy. Section III discusses the principal QoS parameters used to evaluate cloud service performance. Section IV presents a literature review of QoS management techniques in multi-tenant cloud computing. Section V provides a comparative analysis of existing QoS optimization techniques and identifies their strengths and limitations. Section VI identifies major research gaps and discusses emerging research directions. Finally, Section VII concludes the survey.

II. COMMUNITY CLOUD AND MULTI-TENANCY

Cloud computing has revolutionized the delivery of computing services by enabling scalable, on-demand, and cost-effective access to computing resources over the Internet. Based on ownership, governance, and accessibility, cloud infrastructures are categorized into four deployment models: public, private, hybrid, and community clouds [1]. Among these, the community cloud has emerged as an effective deployment model for organizations that share common security policies, regulatory frameworks, and operational objectives. By enabling multiple organizations to utilize a common infrastructure, community clouds improve resource utilization, reduce operational costs, and facilitate secure collaboration. The adoption of multi-tenancy further enhances infrastructure efficiency by allowing multiple tenants to share physical computing resources while maintaining logical isolation through virtualization technologies [2], [3]. However, resource sharing introduces challenges related to Quality of Service (QoS), tenant isolation, resource contention, security, and Service Level Agreement (SLA) compliance, making intelligent resource management essential for sustainable community cloud environments [5], [6].

A. Cloud Deployment Models

The National Institute of Standards and Technology (NIST) classifies cloud computing into four deployment models: public, private, hybrid, and community clouds [1]. Each deployment model is designed to satisfy different organizational requirements concerning infrastructure

ownership, security, regulatory compliance, and resource management.

The **public cloud** is owned and managed by third-party cloud service providers such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform. Computing resources—including virtual machines, storage, databases, and networking services—are delivered over the Internet using a pay-as-you-go model. Public cloud infrastructures provide high scalability, elasticity, and cost efficiency through extensive resource sharing. However, because resources are shared among unrelated organizations, concerns regarding data privacy, regulatory compliance, and predictable performance remain significant challenges, particularly for mission-critical applications [2].

In contrast, the **private cloud** is dedicated to a single organization and provides greater control over infrastructure management, security, and governance. Private cloud infrastructures are commonly adopted by organizations operating in healthcare, banking, and government sectors where regulatory compliance and data confidentiality are essential. Although private clouds offer enhanced customization and security, they require higher deployment and maintenance costs because the infrastructure is not shared with external organizations [3].

The **hybrid cloud** integrates public and private cloud infrastructures to combine the advantages of scalability and security. Organizations typically deploy sensitive applications within private cloud environments while utilizing public cloud resources for workload expansion, disaster recovery, and computationally intensive tasks. Although hybrid cloud architectures improve operational flexibility, maintaining interoperability, workload portability, and consistent QoS across heterogeneous infrastructures remains a complex research challenge [4].

The **community cloud** is specifically designed for organizations that share similar governance policies, security requirements, and regulatory obligations. Typical examples include healthcare institutions, universities, financial organizations, and government agencies that require secure information sharing while complying with common regulatory standards. Unlike public clouds, where users are unrelated, community cloud participants operate under mutually agreed governance frameworks that facilitate trust, collaboration, and secure resource sharing. Consequently, community clouds provide an effective balance between the security of private clouds and the cost efficiency of shared infrastructure [1], [5].

B. Community Cloud Architecture

Community cloud architecture integrates shared computing infrastructure with governance mechanisms to enable secure collaboration among multiple organizations while preserving logical isolation between tenants. The architecture consists of resource pooling, shared infrastructure, governance policies, and security mechanisms that collectively support efficient resource utilization and reliable service delivery.

Resource pooling enables computing resources such as processors, memory, storage devices, and network bandwidth to be dynamically allocated according to workload

requirements. Through virtualization technologies, physical resources are abstracted into logical resource pools that support elastic resource provisioning and efficient workload distribution. This dynamic allocation improves infrastructure utilization and enables cloud providers to accommodate fluctuating application demands without excessive hardware provisioning [2], [3].

A key architectural feature of community clouds is the use of **shared physical infrastructure**, where participating organizations utilize common servers, storage systems, networking equipment, and virtualization platforms. Logical isolation is maintained through hypervisors, virtual machines, and software-defined networking technologies, ensuring operational independence while maximizing resource utilization. Shared infrastructure significantly reduces capital expenditure and operational costs; however, effective scheduling and resource management are essential to minimize resource contention and maintain consistent QoS [3], [6].

Effective **governance** is fundamental to community cloud operation. Governance mechanisms define organizational policies related to authentication, authorization, workload prioritization, resource allocation, auditing, regulatory compliance, and SLA management. These policies ensure accountability, transparency, and equitable resource sharing among participating organizations while supporting common operational objectives [1], [3].

Because multiple organizations operate within a common infrastructure, **security** remains a critical architectural requirement. Identity and Access Management (IAM), Role-Based Access Control (RBAC), multi-factor authentication, encryption, intrusion detection systems, and continuous monitoring are commonly employed to protect tenant data and maintain confidentiality, integrity, and availability. These mechanisms also support compliance with industry-specific regulations and reduce the risk of cross-tenant attacks and unauthorized information disclosure [5].

C. Multi-Tenancy

Multi-tenancy is a fundamental architectural principle that enables multiple independent tenants to share the same computing infrastructure while maintaining logical isolation of applications, services, and data. This approach improves infrastructure utilization, reduces operational expenditure, and enhances scalability by allowing cloud providers to efficiently allocate shared resources among multiple organizations [3].

Multi-tenancy can be implemented using different database architectures depending on application requirements and isolation needs. In the **single database model**, each tenant is assigned an independent database instance, providing maximum isolation and security at the expense of increased infrastructure cost. The **shared database model** allows multiple tenants to utilize a common database management system while maintaining logical separation through independent data structures, thereby improving storage efficiency. The **shared schema model** further increases scalability by storing tenant data within common database tables using unique tenant identifiers, although it requires robust indexing, query optimization, and access control

mechanisms. A **dedicated schema model** represents a compromise between complete database isolation and fully shared databases by assigning each tenant an independent schema within a common database instance, thereby balancing security, scalability, and resource efficiency [3], [4].

The adoption of multi-tenancy significantly reduces infrastructure costs through shared resource utilization while improving elasticity and operational efficiency. Dynamic resource allocation enables idle computing resources to be reallocated among tenants according to workload demands, thereby maximizing infrastructure utilization and supporting rapid application scaling. These capabilities are particularly beneficial in community cloud environments where multiple organizations share common computational resources while maintaining operational independence [2], [3].

Despite these advantages, multi-tenancy introduces several technical challenges. Resource contention caused by simultaneous access to shared processors, memory, storage, and network bandwidth may increase response time and reduce throughput. The **noisy neighbor effect**, in which excessive resource consumption by one tenant negatively affects the performance experienced by others, remains one of the most significant QoS challenges in multi-tenant environments. Furthermore, maintaining tenant isolation, protecting sensitive organizational data, and ensuring equitable resource allocation require sophisticated scheduling algorithms and continuous monitoring [5], [6]. Meeting Service Level Agreement (SLA) requirements therefore depends on intelligent resource management strategies capable of dynamically adapting to changing workload conditions while balancing performance, fairness, and infrastructure utilization.

III. QUALITY OF SERVICE (QOS) IN MULTI-TENANT COMMUNITY CLOUD

Quality of Service (QoS) is a fundamental performance metric that determines the ability of a cloud computing infrastructure to deliver reliable, scalable, and efficient services while satisfying the performance requirements specified in Service Level Agreements (SLAs). In multi-tenant community cloud environments, QoS management becomes particularly challenging because multiple organizations simultaneously share computing resources, including processors, memory, storage, and network bandwidth, while expecting predictable application performance. The dynamic nature of cloud workloads, resource contention among tenants, heterogeneous application requirements, and varying organizational priorities require intelligent resource management mechanisms capable of maintaining service reliability without compromising infrastructure utilization. Consequently, modern community cloud platforms employ adaptive resource allocation, virtualization, workload scheduling, continuous monitoring, and predictive resource management techniques to ensure fair resource distribution and minimize SLA violations [1]–[3].

The National Institute of Standards and Technology (NIST) identifies rapid elasticity, resource pooling, broad network access, measured service, and on-demand self-service as the fundamental characteristics of cloud computing that directly influence QoS performance [1]. These characteristics enable

cloud providers to dynamically provision computing resources according to workload demand while maximizing infrastructure utilization. However, resource sharing also introduces performance variability caused by workload fluctuations, virtualization overhead, and competition among multiple tenants. Armbrust et al. [2] identified scalability, resource contention, performance predictability, and workload management as key research challenges affecting cloud service quality, whereas Buyya et al. [3] emphasized that QoS management should simultaneously optimize application performance, infrastructure utilization, operational cost, and SLA compliance. More recent scheduling frameworks integrate workload prediction, adaptive scheduling algorithms, and intelligent resource allocation strategies to improve fairness, resource utilization, and service reliability in multi-tenant cloud environments [4], [5].

A. QoS Parameters

QoS evaluation in multi-tenant community clouds is based on several performance metrics that collectively measure the efficiency, responsiveness, reliability, and scalability of cloud services. These parameters are interdependent and must be optimized simultaneously to maintain consistent service quality under dynamic workload conditions.

1) Response Time

Response time represents the total time required for a cloud service to receive, process, and return a response to a user request. It includes network transmission delay, queue waiting time, processing time, and resource allocation overhead. Since response time directly influences user experience and application performance, cloud providers employ dynamic scheduling, intelligent load balancing, and adaptive resource provisioning to minimize processing delays while maintaining efficient utilization of shared infrastructure [2], [4].

2) Throughput

Throughput measures the number of computational tasks, service requests, or transactions successfully processed within a specified time interval. Higher throughput indicates efficient utilization of computational resources and greater processing capability. In multi-tenant community clouds, workload scheduling algorithms attempt to maximize throughput while preventing resource contention and ensuring equitable resource allocation among competing tenants [3], [5].

3) Availability and Reliability

Availability represents the percentage of time during which cloud services remain operational and accessible to authorized users, whereas reliability describes the ability of the cloud infrastructure to perform consistently without service interruption or failure over a given period. High availability and reliability are achieved through redundancy, replication, virtual machine migration, checkpointing, automatic failover, and fault-tolerant scheduling mechanisms that minimize service disruption and improve operational continuity [1], [6].

4) Scalability and Elasticity

Scalability refers to the capability of cloud infrastructure to accommodate increasing workloads by expanding computational resources through horizontal or vertical scaling. Elasticity complements scalability by automatically

provisioning and releasing computing resources according to real-time workload variations. Together, these characteristics enable community cloud platforms to efficiently support fluctuating application demands while maintaining consistent QoS and reducing unnecessary infrastructure costs [1]–[3].

5) Latency and Jitter

Latency is the communication delay experienced during data transmission between cloud users and service providers, whereas jitter represents variations in packet transmission delay. These metrics are particularly significant for delay-sensitive applications such as healthcare monitoring, financial transactions, industrial automation, online education, multimedia streaming, and video conferencing. Optimized routing, Software-Defined Networking (SDN), edge computing, and traffic engineering techniques are widely employed to reduce latency and jitter, thereby improving the responsiveness of distributed cloud services [5].

6) Service Level Agreement (SLA) Compliance

Service Level Agreements define contractual commitments between cloud providers and tenants regarding expected performance metrics such as availability, response time, throughput, reliability, and recovery time. Maintaining SLA compliance is one of the primary objectives of QoS management because violations directly affect customer satisfaction, operational reliability, and provider reputation. Continuous performance monitoring, adaptive workload

scheduling, predictive resource allocation, and intelligent orchestration frameworks are commonly employed to minimize SLA violations and maintain agreed service quality [3]–[5].

7) Energy Efficiency and Fairness

Energy efficiency has become an important QoS consideration because modern cloud data centers consume substantial electrical power. Energy-aware resource management techniques dynamically consolidate workloads, optimize virtual machine placement, migrate workloads, and deactivate underutilized resources to reduce power consumption while maintaining acceptable service performance [6]. In addition, fairness is essential in multi-tenant community cloud environments to ensure equitable distribution of shared resources among participating organizations. Fairness-aware scheduling algorithms prevent resource monopolization, mitigate the noisy-neighbor effect, and balance competing workload demands, thereby maintaining consistent QoS across all tenants [4], [5].

B. Comparative Analysis of QoS Parameters

The major QoS parameters and their corresponding optimization objectives are summarized in Table I. Although each metric evaluates a different aspect of cloud performance, they are closely interrelated and collectively determine the effectiveness of QoS management in multi-tenant community cloud environments.

TABLE I
Comparison of Major QoS Parameters in Multi-Tenant Community Cloud

QoS Parameter	Performance Objective	Representative Techniques	Impact on QoS
Response Time	Minimize	Dynamic scheduling, load balancing	Faster request processing
Throughput	Maximize	Parallel execution, workload optimization	Improved resource utilization
Availability	Maximize	Replication, redundancy, failover	Continuous service delivery
Reliability	Maximize	Fault tolerance, VM migration	Improved system stability
Scalability	Maximize	Horizontal and vertical scaling	Supports increasing workloads
Elasticity	Maximize	Auto-scaling, container orchestration	Efficient resource utilization
Latency	Minimize	Edge computing, optimized routing	Improved real-time performance
Jitter	Minimize	SDN, QoS routing, traffic engineering	Stable multimedia communication
SLA Compliance	Maximize	Continuous monitoring, adaptive scheduling	Higher service reliability
Energy Efficiency	Maximize	Energy-aware VM consolidation	Reduced operational cost
Fairness	Maximize	Fair-share scheduling, priority allocation	Balanced resource sharing

The optimization of QoS parameters requires balancing multiple and often conflicting objectives. For example, maximizing throughput may increase response time during peak workloads, while aggressive energy-saving strategies may adversely affect service availability. Consequently, recent QoS management frameworks increasingly adopt multi-objective optimization techniques that simultaneously consider performance, fairness, energy efficiency, and SLA compliance. Such integrated approaches are particularly important in community cloud environments, where multiple organizations share common infrastructure while expecting predictable and reliable service quality [4]–[6].

IV. LITERATURE REVIEW

Quality of Service (QoS) management in multi-tenant cloud computing has evolved from static resource allocation approaches to intelligent, adaptive, and autonomous resource management frameworks that address workload heterogeneity, tenant fairness, Service Level Agreement (SLA) compliance, scalability, and energy efficiency. Recent advances in virtualization, containerization, cloud-native technologies, artificial intelligence (AI), and edge computing have significantly improved the ability of cloud providers to deliver reliable and scalable services while efficiently utilizing shared infrastructure. This section presents a comprehensive review of

the major contributions that have shaped QoS-aware resource management in multi-tenant community cloud environments.

A. Cloud Computing Foundation and Community Cloud Evolution

The conceptual foundation of cloud computing was established by Mell and Grance [1], who introduced the widely accepted NIST definition of cloud computing. Their work identified the five essential characteristics of cloud computing, namely on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service, together with the four deployment models including the community cloud. Although QoS optimization was not explicitly addressed, the concepts of resource pooling and rapid elasticity laid the groundwork for dynamic resource provisioning and adaptive QoS management in modern cloud environments.

Building upon the NIST framework, Armbrust et al. [2] provided one of the earliest comprehensive studies of cloud computing by discussing its economic advantages and technical capabilities. Their work identified several research challenges, including performance predictability, scalability, resource scheduling, security, and data management. The authors emphasized that efficient resource allocation and intelligent scheduling mechanisms are essential for satisfying diverse QoS requirements while maximizing infrastructure utilization. Similarly, Buyya et al. [3] proposed a comprehensive cloud computing architecture integrating virtualization, resource provisioning, and SLA management. Their framework demonstrated how virtualization enables multiple tenants to share computing resources while maintaining logical isolation and supporting dynamic resource allocation. These studies collectively established the architectural principles that underpin QoS-aware community cloud infrastructures.

Vaquero et al. [4] further investigated the evolution of cloud computing by examining virtualization technologies and service-oriented computing models. Their analysis highlighted virtualization as the enabling technology for efficient resource sharing, workload migration, and infrastructure flexibility. However, the authors also observed that increasing tenant density may introduce resource contention and performance variability, thereby emphasizing the need for adaptive QoS-aware scheduling mechanisms.

B. QoS-Aware Resource Management and Scheduling

Resource scheduling remains one of the most critical aspects of QoS management in multi-tenant cloud environments. Zhang et al. [5] identified resource management, scalability, interoperability, energy efficiency, and QoS as major research challenges in cloud computing. They argued that scheduling algorithms should optimize multiple QoS objectives simultaneously, including response time, throughput, resource utilization, and SLA compliance, rather than focusing on a single performance metric.

Energy-efficient resource management has also received significant attention because cloud data centers consume substantial computational power. Beloglazov and Buyya [6] proposed adaptive virtual machine consolidation techniques

that dynamically migrate virtual machines based on workload conditions to reduce energy consumption while maintaining acceptable QoS. Their experimental results demonstrated that intelligent VM consolidation significantly decreases power consumption with minimal SLA violations, establishing an important relationship between sustainability and QoS optimization.

Focusing specifically on multi-tenant environments, Jia et al. [7] conducted a systematic review of scheduling approaches for multi-tenancy cloud platforms. After analyzing more than fifty primary studies, the authors concluded that effective scheduling frameworks must jointly optimize computing resources, storage allocation, and QoS requirements instead of considering these objectives independently. They also identified heterogeneous resource management, GPU scheduling, and intelligent scheduling algorithms as promising research directions.

Ru et al. [8] proposed a deadline-constrained and data-locality-aware scheduling framework that integrates adaptive resource allocation, fairness-aware scheduling, and data locality optimization. Their approach significantly improved resource utilization, system throughput, and job completion time while satisfying tenant-specific QoS constraints. Likewise, Hilman et al. [9] investigated Workflow-as-a-Service (WaaS) platforms and proposed an elastic budget-constrained scheduling algorithm for multi-tenant scientific cloud environments. Their work demonstrated that intelligent resource-sharing policies improve workflow completion time, cost efficiency, and overall infrastructure utilization while meeting user-defined QoS requirements.

C. Virtualization, Containerization, and Cloud-Native Resource Management

The transition from virtual machine-based infrastructures to container-based cloud-native environments has substantially improved QoS management in multi-tenant clouds. Bernstein [10] demonstrated that container technologies such as Docker provide lightweight virtualization with lower startup latency, reduced computational overhead, and improved application portability compared with traditional virtual machines. Containerization enables cloud providers to support higher tenant density while maintaining acceptable QoS and efficient resource utilization.

Extending this concept, Burns et al. [11] described the evolution of Google's cluster management platforms from Borg and Omega to Kubernetes. Kubernetes automates container deployment, workload scheduling, fault recovery, and elastic scaling using declarative resource management principles. Its self-healing and auto-scaling capabilities significantly improve service availability, scalability, and operational efficiency, making Kubernetes a key orchestration platform for QoS-aware resource management in modern community cloud infrastructures.

D. Performance Variability and Intelligent Cloud Management

Maintaining predictable application performance remains challenging in multi-tenant cloud environments because of workload diversity and resource contention. Leitner and Cito [12] conducted an extensive empirical analysis of

Infrastructure-as-a-Service (IaaS) cloud platforms and demonstrated that application performance varies considerably because of hardware heterogeneity, virtualization overhead, and tenant interference. Their findings emphasized the importance of continuous workload monitoring and adaptive resource allocation to achieve consistent QoS.

To address these limitations, Buyya, Calheiros, and Beloglazov [13] introduced an autonomic cloud computing architecture capable of self-monitoring, self-configuration, self-optimization, and self-healing. Their framework automatically adjusts resource allocation according to workload characteristics and SLA requirements without human intervention, thereby improving infrastructure utilization while minimizing SLA violations and operational costs. This work laid the foundation for intelligent and autonomous cloud resource management.

Artificial intelligence has further expanded the capabilities of QoS optimization. Although LeCun, Bengio, and Hinton [15] primarily introduced deep learning as a general machine learning paradigm, its ability to learn complex patterns from large datasets has enabled numerous cloud computing applications, including workload prediction, anomaly detection, intelligent scheduling, and resource demand forecasting. AI-driven resource management frameworks increasingly employ deep neural networks to improve predictive resource allocation and adaptive QoS optimization.

E. Emerging QoS Optimization Technologies

Recent cloud computing research has increasingly focused on integrating edge computing and sustainable resource management into cloud infrastructures. Shi and Dustdar [16] proposed extending cloud services toward the network edge to reduce latency, communication overhead, and bandwidth consumption for delay-sensitive applications. Their work demonstrated that hybrid edge-cloud architectures significantly improve QoS by enabling faster response times and localized data processing.

In parallel, Beloglazov, Abawajy, and Buyya [17] developed heuristic algorithms for energy-aware virtual machine placement and workload consolidation in cloud data centers. Their proposed heuristics dynamically allocate resources according to workload fluctuations, reducing power consumption while maintaining SLA compliance and acceptable QoS levels. This research highlights the growing importance of balancing energy efficiency, environmental sustainability, and service quality in next-generation cloud infrastructures.

V. COMPARATIVE ANALYSIS OF QOS TECHNIQUES IN MULTI-TENANT COMMUNITY CLOUD

Quality of Service (QoS) management in multi-tenant community cloud computing has evolved from conventional virtualization-based resource allocation to intelligent and adaptive resource management frameworks. Early cloud research primarily emphasized virtualization, resource pooling, and infrastructure provisioning to improve resource utilization and service availability [1]–[3]. With the emergence of cloud-native applications, container technologies, Software-Defined Networking (SDN), and Artificial Intelligence (AI), recent studies have focused on optimizing multiple QoS objectives simultaneously, including response time, throughput, latency, scalability, energy efficiency, fairness, and Service Level Agreement (SLA) compliance [6]–[17]. Consequently, QoS optimization has become a multi-objective problem that requires balancing resource utilization, application performance, operational cost, and tenant satisfaction.

Table II summarizes representative QoS optimization techniques proposed in the literature and compares their primary methodologies, QoS metrics, strengths, and limitations. The comparison indicates that although substantial progress has been achieved in cloud resource management, no existing technique simultaneously satisfies all QoS requirements in multi-tenant community cloud environments.

TABLE II
Comparative Analysis of QoS Techniques in Multi-Tenant Community Cloud

Ref.	Technique	QoS Metrics	Major Contribution	Limitation
[1]	NIST Cloud Architecture	Availability, Scalability	Standardized cloud characteristics and deployment models	No QoS optimization framework
[2]	Cloud Resource Management	Scalability, Response Time	Identified major cloud computing challenges	Conceptual study without scheduling implementation
[3]	Dynamic Resource Provisioning	Throughput, SLA, Resource Utilization	Virtualization-based resource allocation	Limited fairness consideration
[4]	Virtualization Framework	Scalability, Resource Sharing	Defined virtualization as the cloud foundation	Limited QoS scheduling discussion
[5]	Cloud Computing Survey	Response Time, Throughput	Comprehensive cloud research challenges	No implementation framework
[6]	Energy-Aware VM Consolidation	Energy Efficiency, SLA	Reduced power consumption with QoS preservation	VM-centric approach
[7]	Multi-Tenant Scheduling Survey	Response Time, Throughput, Fairness	Comprehensive classification of scheduling methods	No new scheduling algorithm
[8]	Deadline-Constrained Scheduling	Response Time, Throughput, Data Locality	Improved scheduling efficiency and SLA compliance	Limited heterogeneous cloud evaluation

Ref.	Technique	QoS Metrics	Major Contribution	Limitation
[9]	Workflow Resource Sharing	Cost, Throughput	Efficient workflow scheduling	Limited to scientific workflows
[10]	Container-Based Virtualization	Scalability, Elasticity	Lightweight virtualization	Security and orchestration issues
[11]	Kubernetes Orchestration	Availability, Reliability	Automated deployment and scaling	Default scheduler ignores tenant priority
[12]	Performance Variability Analysis	Response Time, Latency	Identified cloud performance variability	No adaptive scheduling mechanism
[13]	Autonomic Cloud Computing	SLA, Resource Utilization	Self-managing cloud architecture	Limited AI integration
[14]	Containerization for PaaS	Elasticity, Scalability	Improved portability and deployment	No workload prioritization
[15]	Deep Learning	Prediction Accuracy	Foundation for intelligent prediction models	Not cloud-specific
[16]	Edge Computing	Latency, Availability	Reduced communication delay	Additional infrastructure complexity
[17]	Energy-Aware Scheduling	Energy Efficiency, Throughput	Sustainable cloud resource management	Focuses mainly on energy optimization

A. Comparative Evaluation

Resource allocation remains the cornerstone of QoS management because efficient distribution of computing resources directly affects throughput, response time, and SLA compliance. Early cloud architectures relied on virtualization to improve infrastructure utilization through dynamic resource provisioning, thereby enabling multiple tenants to share physical resources while maintaining logical isolation [3], [4]. Although these approaches significantly enhanced resource efficiency, they primarily considered CPU and memory utilization without explicitly addressing tenant fairness or workload diversity.

Scheduling algorithms have subsequently become the primary mechanism for improving QoS in multi-tenant cloud environments. The systematic review conducted by Jia et al. demonstrated that existing scheduling techniques generally optimize individual QoS metrics such as execution time, throughput, or resource utilization rather than multiple objectives simultaneously [7]. Ru et al. addressed this limitation by integrating deadline constraints, adaptive scheduling, and data locality awareness, thereby improving workload completion time, throughput, and SLA compliance in multi-tenant cloud environments [8].

Containerization has further transformed cloud resource management by replacing heavyweight virtual machines with lightweight containers. Bernstein demonstrated that containers provide faster deployment, lower resource overhead, and greater application portability than conventional virtualization technologies [10]. Similarly, Pahl showed that container-based Platform-as-a-Service (PaaS) architectures improve scalability, elasticity, and deployment efficiency while simplifying application management [14]. Despite these advantages, ensuring secure container isolation and fair resource sharing among tenants remains a significant challenge.

Container orchestration platforms such as Kubernetes have introduced automated deployment, self-healing, load balancing, and auto-scaling capabilities for cloud-native applications [11]. Kubernetes significantly improves service

availability and infrastructure utilization through declarative resource management; however, its default scheduler primarily considers resource availability rather than tenant priority, workload criticality, or organizational policies. Consequently, integrating intelligent scheduling algorithms with Kubernetes remains an important research direction for QoS-aware community cloud environments [11].

Artificial Intelligence has recently emerged as an enabling technology for predictive QoS management. Deep learning techniques provide accurate workload prediction and anomaly detection capabilities that enable proactive rather than reactive resource allocation [15]. AI-driven scheduling algorithms can continuously analyze workload behavior and optimize resource allocation to reduce SLA violations while improving response time and infrastructure utilization. Nevertheless, the computational complexity and limited interpretability of many AI models continue to hinder their practical deployment in production cloud systems.

Edge computing represents another important advancement in QoS optimization by extending computational resources closer to end users. Shi and Dustdar demonstrated that edge-enabled cloud architectures significantly reduce communication latency and bandwidth consumption for delay-sensitive applications, thereby improving service responsiveness and user experience [16]. Hybrid edge-cloud architectures are therefore increasingly adopted for applications requiring real-time processing, including healthcare monitoring, industrial automation, and intelligent transportation systems.

Energy efficiency has also become a critical QoS consideration due to the growing operational costs and environmental impact of cloud data centers. Beloglazov and Buyya proposed energy-aware virtual machine consolidation techniques that dynamically migrate workloads to reduce power consumption while maintaining acceptable QoS levels [6]. Their subsequent work further demonstrated that intelligent workload placement and energy-aware scheduling

algorithms can simultaneously improve infrastructure utilization and minimize SLA violations [17].

VI. RESEARCH GAPS AND FUTURE RESEARCH DIRECTIONS

Although considerable progress has been achieved in Quality of Service (QoS) management for cloud computing through virtualization, dynamic resource provisioning, intelligent scheduling, containerization, and cloud orchestration, several technical challenges remain unresolved in multi-tenant community cloud environments. Most existing QoS optimization techniques have been developed for public or private cloud infrastructures, whereas comparatively limited attention has been devoted to community clouds that support multiple organizations operating under common governance, security policies, and regulatory requirements. Furthermore, many existing approaches optimize only a limited subset of QoS parameters without simultaneously considering tenant fairness, workload heterogeneity, sustainability, and intelligent decision-making. Therefore, developing comprehensive QoS management frameworks that address these limitations remains an important research direction [3], [5], [7].

A. Adaptive Resource Scheduling

Most scheduling algorithms reported in the literature employ heuristic or rule-based policies that remain static throughout execution. Although such algorithms provide low computational overhead, they cannot efficiently adapt to rapidly changing workloads, resulting in resource underutilization, increased response time, and Service Level Agreement (SLA) violations during peak demand [7], [8]. Modern cloud infrastructures require adaptive scheduling mechanisms capable of continuously monitoring workload characteristics and dynamically reallocating computational resources according to real-time demand. Artificial Intelligence (AI), reinforcement learning, and predictive analytics provide promising solutions for designing self-adaptive scheduling frameworks that optimize multiple QoS objectives simultaneously [13], [15].

B. Fairness-Aware Multi-Tenant Resource Allocation

Fair resource allocation remains one of the least investigated QoS objectives in multi-tenant community cloud computing. Most existing resource allocation techniques primarily optimize throughput, execution time, or infrastructure utilization while overlooking equitable resource sharing among participating tenants [3], [7]. Consequently, tenants with higher computational demands may monopolize shared resources, leading to the noisy-neighbor effect, performance degradation, and inconsistent QoS for other organizations. Future scheduling frameworks should incorporate fairness-aware policies that consider tenant priorities, historical resource consumption, organizational requirements, and workload diversity while maintaining overall system efficiency [7], [8].

C. Intelligent Workload Prediction

Current cloud resource management approaches are predominantly reactive, allocating resources only after workload changes have occurred. Such reactive mechanisms often lead to delayed scaling decisions, temporary performance

degradation, and unnecessary SLA violations [6], [13]. Although machine learning techniques have recently been introduced for workload prediction, existing models exhibit limited accuracy under highly dynamic and heterogeneous cloud workloads. Deep learning and reinforcement learning techniques can improve prediction accuracy by analyzing historical workload patterns and proactively provisioning computing resources before performance degradation occurs [15]. Integrating predictive resource management with cloud orchestration platforms remains an important research challenge.

D. QoS Management for Community Cloud Environments

A significant proportion of cloud QoS research has concentrated on public, private, and hybrid cloud infrastructures, whereas comparatively fewer studies have addressed the distinctive characteristics of community cloud computing [1], [5]. Community clouds require multiple independent organizations to share computational resources while complying with common governance policies, regulatory requirements, and security constraints. Consequently, QoS optimization in community cloud environments must simultaneously satisfy tenant-specific priorities, collaborative resource sharing, regulatory compliance, and service fairness. Developing QoS frameworks specifically designed for community cloud architectures therefore represents a significant research opportunity.

E. Continuous SLA Monitoring and Self-Healing

Service Level Agreements define contractual QoS commitments concerning response time, availability, reliability, throughput, and security. Existing scheduling algorithms primarily attempt to reduce SLA violations through efficient resource allocation; however, many approaches identify violations only after performance degradation has already occurred [6], [8]. Future cloud management systems should incorporate continuous SLA monitoring, predictive violation detection, automated workload migration, and self-healing mechanisms capable of maintaining contractual QoS guarantees under highly dynamic workload conditions [13]. Such autonomous management systems would significantly improve cloud reliability while reducing operational overhead.

F. Explainable Artificial Intelligence for Cloud Management

Artificial Intelligence has become increasingly important for workload prediction, anomaly detection, and autonomous resource scheduling in cloud computing [15]. Nevertheless, many AI-based scheduling frameworks operate as black-box models whose decision-making processes cannot be easily interpreted by cloud administrators. This lack of transparency limits trust and accountability, particularly in community cloud environments involving multiple organizations with different governance policies. Explainable Artificial Intelligence (XAI) offers an emerging solution by providing interpretable scheduling decisions, thereby improving transparency, regulatory compliance, and administrator confidence in AI-assisted cloud management systems.

G. Sustainable and Energy-Efficient Cloud Computing

The increasing size of cloud data centers has resulted in significant energy consumption and operational costs.

Although energy-aware virtual machine consolidation techniques have demonstrated considerable reductions in power consumption, most existing QoS optimization approaches continue to prioritize application performance without simultaneously considering environmental sustainability [6], [17]. Future research should investigate multi-objective optimization algorithms capable of balancing response time, throughput, availability, energy consumption, carbon emissions, and resource utilization. Integrating renewable energy awareness and carbon-efficient scheduling into cloud orchestration frameworks represents a promising direction for sustainable community cloud computing.

H. Integrated QoS Orchestration for Cloud-Native Platforms

Current cloud management platforms generally optimize individual QoS metrics such as response time, throughput, or

energy efficiency independently. Practical community cloud environments, however, require simultaneous optimization of multiple interdependent QoS objectives, including latency, scalability, reliability, fairness, SLA compliance, security, and sustainability [5], [7]. Emerging technologies such as Kubernetes, Software-Defined Networking (SDN), edge computing, and AI-based orchestration provide opportunities to develop unified QoS management frameworks capable of continuously monitoring infrastructure conditions and dynamically adapting resource allocation policies [10], [11], [16]. Such integrated orchestration platforms are expected to constitute the next generation of intelligent community cloud management systems.

TABLE III

Summary of Research Gaps and Future Research Directions

Research Gap	Current Limitation	Future Research Direction
Static resource scheduling	Fixed heuristic scheduling cannot adapt to workload fluctuations	AI-driven adaptive scheduling and reinforcement learning
Limited tenant fairness	Unequal resource allocation causes noisy-neighbor effects	Fairness-aware multi-objective scheduling algorithms
Reactive resource management	Resources allocated after workload changes	Predictive workload forecasting using AI and deep learning
Community cloud QoS	Most studies focus on public/private clouds	QoS frameworks specifically designed for community cloud environments
Weak SLA enforcement	SLA violations detected after occurrence	Continuous SLA monitoring and self-healing architectures
Limited AI transparency	Black-box scheduling decisions	Explainable AI (XAI)-based cloud orchestration
High energy consumption	Performance prioritized over sustainability	Green computing and carbon-aware resource scheduling
Single-objective QoS optimization	Independent optimization of QoS metrics	Integrated multi-objective QoS orchestration using Kubernetes, SDN, Edge Computing, and AI

VII. CONCLUSION

Quality of Service (QoS) has become a fundamental requirement for ensuring efficient, reliable, and scalable service delivery in multi-tenant community cloud computing. The increasing adoption of community cloud infrastructures by organizations with shared security, governance, and regulatory requirements has highlighted the need for effective QoS management that balances resource utilization with consistent service performance. Unlike conventional cloud environments, community clouds must satisfy diverse tenant requirements while maintaining fairness, security, scalability, and compliance with Service Level Agreements (SLAs).

This survey presented a comprehensive review of QoS management in multi-tenant community cloud environments. It examined the concepts of community cloud computing and multi-tenancy, discussed the architectural characteristics and deployment models, and analyzed the major QoS parameters that influence cloud performance, including response time, throughput, availability, reliability, scalability, elasticity, latency, energy efficiency, fairness, and SLA compliance. In addition, the survey reviewed the evolution of QoS optimization techniques, ranging from virtualization-based

resource management to modern cloud-native technologies such as containerization, Kubernetes orchestration, Software-Defined Networking (SDN), edge computing, and Artificial Intelligence (AI)-based resource scheduling.

The comparative analysis demonstrated that existing QoS optimization approaches have significantly improved cloud resource management through dynamic scheduling, intelligent resource allocation, workload prediction, and energy-aware computing. However, most current solutions focus on optimizing individual QoS metrics rather than addressing multiple performance objectives simultaneously. Challenges such as tenant fairness, heterogeneous workload management, dynamic resource provisioning, explainable AI, continuous SLA enforcement, and sustainable cloud operation remain open research problems, particularly in community cloud environments where multiple organizations share common infrastructure.

Future research should focus on developing integrated, intelligent, and autonomous QoS management frameworks capable of simultaneously optimizing performance, scalability, reliability, fairness, security, energy efficiency, and regulatory compliance. The integration of Artificial Intelligence, Machine

Learning, Reinforcement Learning, Explainable Artificial Intelligence, Kubernetes orchestration, Software-Defined Networking, edge computing, and predictive analytics offers significant potential for building next-generation cloud platforms that can proactively adapt to changing workload conditions and tenant requirements.

In conclusion, QoS management continues to evolve from traditional resource provisioning toward intelligent and self-adaptive cloud orchestration. Although substantial progress has been achieved in cloud resource management and service optimization, further research is required to develop comprehensive multi-objective QoS frameworks that effectively address the complex requirements of multi-tenant community cloud environments. Such advancements will play a crucial role in improving the reliability, efficiency, sustainability, and overall performance of future community cloud infrastructures.

REFERENCES

- [1] P. Mell and T. Grance, The NIST Definition of Cloud Computing, NIST Special Publication 800-145, National Institute of Standards and Technology, Gaithersburg, MD, USA, Sep. 2011.
- [2] M. Armbrust et al., "A View of Cloud Computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010.
- [3] R. Buyya, J. Broberg, and A. Goscinski, *Cloud Computing: Principles and Paradigms*. Hoboken, NJ, USA: John Wiley & Sons, 2011.
- [4] L. M. Vaquero, L. Roderio-Merino, J. Cáceres, and M. Lindner, "A Break in the Clouds: Towards a Cloud Definition," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 50–55, Jan. 2009.
- [5] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud Computing: State-of-the-Art and Research Challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, May 2010.
- [6] A. Beloglazov and R. Buyya, "Energy Efficient Resource Management in Virtualized Cloud Data Centers," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, May 2012.
- [7] R. Jia, Y. Yang, J. Grundy, J. Keung, and L. Hao, "A Systematic Review of Scheduling Approaches on Multi-Tenancy Cloud Platforms," *Information and Software Technology*, vol. 132, Art. no. 106478, Apr. 2021.
- [8] J. Ru, Y. Yang, J. Grundy, J. Keung, and L. Hao, "An Efficient Deadline-Constrained and Data Locality-Aware Dynamic Scheduling Framework for Multitenancy Clouds," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 5, Art. no. e6037, Mar. 2021.
- [9] M. H. Hilman, M. A. Rodriguez, and R. Buyya, "Resource-Sharing Policy in Multi-Tenant Scientific Workflow-as-a-Service Cloud Platform," *Future Generation Computer Systems*, vol. 99, pp. 549–561, Oct. 2019.
- [10] D. Bernstein, "Containers and Cloud: From LXC to Docker to Kubernetes," *IEEE Cloud Computing*, vol. 1, no. 3, pp. 81–84, Sep.–Oct. 2014.
- [11] B. Burns, B. Grant, D. Oppenheimer, E. Brewer, and J. Wilkes, "Borg, Omega, and Kubernetes," *Communications of the ACM*, vol. 59, no. 5, pp. 50–57, May 2016.
- [12] P. Leitner and J. Cito, "Patterns in the Chaos: A Study of Performance Variation and Predictability in Public IaaS Clouds," *ACM Transactions on Internet Technology*, vol. 16, no. 3, Art. no. 15, pp. 1–23, Apr. 2016.
- [13] R. Buyya, R. N. Calheiros, and A. Beloglazov, "Autonomic Cloud Computing: Open Challenges and Architectural Elements," in *Proc. Int. Conf. Emerging Applications of Information Technology (EAIT)*, Kolkata, India, 2011, pp. 3–10.
- [14] C. Pahl, "Containerization and the PaaS Cloud," *IEEE Cloud Computing*, vol. 2, no. 3, pp. 24–31, May–Jun. 2015.
- [15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
- [16] W. Shi and S. Dustdar, "The Promise of Edge Computing," *Computer*, vol. 49, no. 5, pp. 78–81, May 2016.
- [17] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, May 2012.