Received: 5 March 2018 | Revised: 8 January 2019 | Accepted: 16 January 2019
DOI: 10.1002/cpe.5186

WILEY

RESEARCH ARTICLE

Applying queue theory for modeling of cloud computing: A systematic review

Einollah Jafarnejad Ghomi[1] | Amir Masoud Rahmani[1] | Nooruldeen Nasih Qader[2,3]

[1]Science and Research Branch, Islamic Azad University, Tehran, Iran
[2]Department of Computer Science, University of Human Development, Sulaymaniyah, Iraq
[3]Department of Computer Science, University of Sulaimani, Sulaymaniyah, Iraq

Correspondence
Amir Masoud Rahmani, Science and Research Branch, Islamic Azad University, Tehran, Iran.
Email: Rahmani@srbiau.ac.ir

Summary

The cloud computing paradigm is an important service in the Internet for sharing and providing resources in a cost-efficient way. Modeling of a cloud system is not an easy task because of the complexity and large scale of such systems. Cloud reliability could be improved by modeling the various aspects of cloud systems, including scheduling, service time, wait time, and hardware and software failures. The aim of this study is to survey research studies done on the modeling of cloud computing using the queuing system in order to identify where more emphasis should be placed in both current and future research directions. This paper follows the goal by investigating the articles published between 2008 and January 2017 in journals and conferences. A systematic mapping study combined with a systematic literature review was performed to find the related literature, and 71 articles were selected as primary studies that were classified in relation to the focus, research type, and contribution type. We classified the modeling techniques of cloud computing using the queuing theory in seven categories based on their focus area: (1) performance, (2) quality of service, (3) workflow scheduling, (4) energy savings, (5) resource management, (6) priority-based servicing, and (7) reliability. A majority of the primary articles focus on performance (37%), 15% of them focus on resource management, 14% of them focus on quality of service, 13% of them focus on workflow scheduling, 13% of them focus on energy savings, 4% of them focus on priority-based servicing for requests, and 4% of them focus on reliability. This work summarizes and classifies the research efforts conducted on applying queue theory for modeling of cloud computing (AQTMCC), providing a good starting point for further research in this area.

KEYWORDS

cloud computing, cloud modeling, queuing systems, queuing theory, systematic review

1 | INTRODUCTION

Cloud computing is the recent evolution in the information technology and is growing very fast due to the extensive use of mobile devices, such as PDAs, cell phones, and tablets.[1] To emphasize the goals of cloud computing, we refer to the definition of NIST (National Institute of Standards and Technology); a cloud model is composed of five characteristics, three service models, and four deployment models.[2] Service models of cloud computing are as follows: (1) Software as a Service (SaaS), such as CRM (Customer Relationship Management), virtual desktop, communications, and games; (2) Platform as a Service (PaaS), such as databases, web servers, and deployment tools; and (3) Infrastructure as a Service (IaaS), such as virtual machines (VMs), storages, servers, and load balancers. Deployment models include private clouds, public clouds, community clouds, and hybrid clouds. The characteristics of a cloud are on-demand self-servicing, elasticity, broad network access, resource pooling, and measured service.[1] A data center in cloud computing consists of several servers that are organized in racks and connected through communication devices, such as routers and switches. As this network could have severe influences on the performance and throughput of applications in such a distributed environment, designers should provide a proper plan for that. Scalability and elasticity properties should also be considered.[3]

Queuing theory is used when you have a lot of jobs, limited resources, and, as a result, long queues and delays. In other words, queuing theory applies anywhere that queues come up.[4] For example, a server in a data center receives service requests from clients and should serve them. The requests enter a queue and will be served based on a service discipline. The goals of applying the queuing theory are twofold: (1) *Predicting the*

system performance. Typically, this means predicting the average delay or delay variability or the probability that delay exceeds some service-level agreement (SLA). However, it also means predicting the number of jobs that will be queuing or the average number of servers being utilized, or any other such metric. (2) *Finding a superlative system design* to improve the performance that takes the form of capacity planning, which is an important topic in cloud environments. By analyzing the queues, one can understand the behavior of the underlying model. A mathematical analysis of models produces formulas that measure system performance metrics, such as average wait time, server utilization, throughput, the probability of exceeding buffer, the distribution of waiting time, the period of server activity, etc.[5]

After conducting a trial survey on the topic of applying queue theory for modeling of cloud computing (AQTMCC), we found that this is an emerging research area among several research fields, such as (1) performance, (2) quality of service, (3) workflow scheduling, (4) energy savings, (5) resource management, (6) priority-based servicing, and (7) reliability. Therefore, in order to provide a true picture of the research studies done so far in the field of AQTMCC, the good practices from systematic mapping studies (SMSs)[6] and systematic literature reviews (SLRs)[7] were combined in this study. Due to the importance of AQTMCC, in this paper, a comprehensive body of SMSs or SLRs to review and analyze is presented. We extracted and synthesized data from the primary studies of AQTMCC to answer our research questions (RQs). At the end, the key contributions of this work are our answers to RQs. As a part of this study, we defined the inclusion and exclusion criteria of relevant primary and secondary studies and systematically developed and refined a systematic map or classification schema of all the selected studies.

As another field of research in AQTMCC, the auto-scaling property in a cloud environment can be mentioned. Qu et al[8] provided a taxonomy of auto-scaling according to the identified challenges and key properties for web application in cloud computing. Auto-scaling allows cloud users to acquire or release computing resources on-demand, which enables web application providers to automatically scale the resources provisioned to their applications without human intervention under a dynamic workload to minimize resource cost while satisfying quality-of-service (QoS) requirements. They mentioned the application of queue in auto-scaling. Chen et al[9] explored the state-of-the-art research studies on a self-aware and self-adaptive cloud auto-scaling system and provided a comprehensive taxonomy. They found that the current research studies on the self-aware and self-adaptive cloud auto-scaling system often require sophisticated designs in different highest-level logical aspects of the auto-scaling engine and mentioned them. Aslanpour et al[10] proposed an executor for the cost-aware auto-scaling mechanism, Suprex, that benefits two heuristic features. Suprex overcomes the challenge of delayed startup for new virtual machines without the help of cool-down time or vertical scaling.

The rest of this paper is structured as follows. Section 2 provides a background review of cloud computing and queuing theory. Section 3 presents our approach to conducting this study. Section 4 contains our classification schemes for the primary AQTMCC studies and other criteria for supporting the data extraction and comparison among these primary studies. In Section 5, we answer RQs and discuss the key results. Finally, the conclusions are presented in Section 6.

2 | BACKGROUND AND RELATED SURVEYS

In this section, we provide some background concepts that are used throughout this paper, including types of literature review, characteristics of cloud computing, data centers, and queuing theory. First, in Section 2.1, we describe the types of literature review. In Section 2.2, we review the basic concepts and the related terminology to AQTMCC. Section 2.3 looks at the data center architecture as an important component of cloud computing. In Section 2.4, we explain further the concepts of queuing theory. Section 2.5 presents the related surveys.

2.1 | Types of literature review

In this subsection, we describe briefly the types of literature review, which are considered secondary studies. According to other works,[6,7,11-15] regular survey, SMS, and SLR are three types of secondary studies; a secondary study is a study that reviews all primary studies for answering a certain RQ. Regular surveys are common and have been done nearly in all topics. Although SMS and SLR share some features, for example, in searching and selecting studies, their goals and analysis approaches are different. SMSs are primarily concerned with structuring a research area, whereas SLRs focus on synthesizing the evidence.[6] In a regular survey, a seminal paper of the specific topic is chosen and looks for papers where this paper is cited or papers that cited that paper. In SLR methodology, a literature review is driven by some very specific research questions that can be answered by empirical research. The identification of appropriate studies, including activities such as searching and inclusion/exclusion, is driven by research questions. Furthermore, it informs the data extraction process applied to each included primary study. SMS methodology provides an overview of a research area; identifies the amount of work, the type of research, and results available; and maps the frequencies of publication over time to see trends. In both methods, SLR and SMS, a researcher should provide some RQs and try to answer them. The objective of RQs in SMSs is to discover research trends; they find the topics covered in the literature and their trends and venues over time. On the other hand, in SLRs, the goal is to find out the evidence, and therefore, a very specific objective has to be formulated.

2.2 | Basic concepts and the related terminology

In this section, we introduce basic concepts and the related terminology, which are used in this paper.[16-20]

Throughput: It is the number of requests that complete their execution per unit time.

Response time: It is the amount of time it takes between submitting a request until the first response is produced.

Scalability: It is the ability of the software system to manage increasing complexity given additional resources. Cloud computing requires scalability with large data set operations.

Quality of service (QoS): It provides a guarantee of the aspects of service quality, such as performance, availability, security, reliability, dependability, and usability. QoS requirements are associated with service providers and end users.

Fault tolerance: It keeps the systems operating even if some of its components are failing. In general, fault tolerance requires fault isolation to the failing components and the availability of reversion mode. Fault-tolerant systems are characterized in terms of outages.

Performance: It is system efficiency with indicators such as response time, waiting time, the probability of task blocking, the probability of immediate service, and the mean number of tasks in the system.

Energy savings: It is the amount of energy saved due to using the queuing theory for cloud computing.

Carbon emission: It is the amount of carbon produced by all resources in cloud computing.

Service time: It is the time required for customer servicing.

Wait time: It is the time that a customer waits in the queue; this excludes the time that the customer spends in the service.

Delay: It is the time representing the total delay in the system; this includes the time that a customer waits in the queue and in the service.

2.3 | Data center in cloud computing

Cloud computing is made up of the applications delivered as services over the Internet and the data centers (the hardware and systems software) that provide those services.[19] A data center, which accomplishes the computation power and storage of the system, is a key component in cloud computing and contains thousands of devices, such as servers, switches, and routers. Proper designing of a data center is critical because it is responsible for handling service requests.[3,21] As stated in the work of Kitchenham et al,[13] the service requests are processed by VMs, which share processing power on the data center's servers. A virtual machine monitor (VMM) is responsible for managing VMs, including provisioning of a server to a VM, VM creation, VM destruction, and VM migration. Servers in a data center are packed into racks. Based on the literature review, we can picture the structure of a cloud data center in Figure 1. As we can see in the Figure, a data center could be viewed in three layers. The first layer manages traffic into and out of the data center. The second layer usually provides important functions, such as domain services, location services, server load balancing, and more. The third layer is where the servers in racks are physically connected to the network. There are typically 20 to 40 servers per rack.[3]

2.4 | Queuing models

Modeling is the process of developing a model from a target, such as a cloud system. In the modeling process, instead of an exact system, experiments are done on a simulation or mathematical model, such as a queuing model. As mentioned, in this paper, we review the literature that paid attention to queuing theory for modeling of cloud environments. Queuing models are classified into two categories: (1) single-queue models, such as M/M/1, G/M/1, and M/M/k, and (2) queuing network models, such as a Jackson network and an open/closed network.[4,22-24] Based on the literature review, some queuing models are shown in Figure 2. In Figure 2A, we can see a simple queue structure that consists of a queue for store arrivals and a server for servicing the arrivals. The arrival rate is λ, the service rate is μ, and the service discipline is FCFS (first come, first serve). A closed-batch system is shown in Figure 2B, an open queuing system is shown in Figure 2C, and a simple Jackson network is shown in Figure 2D.

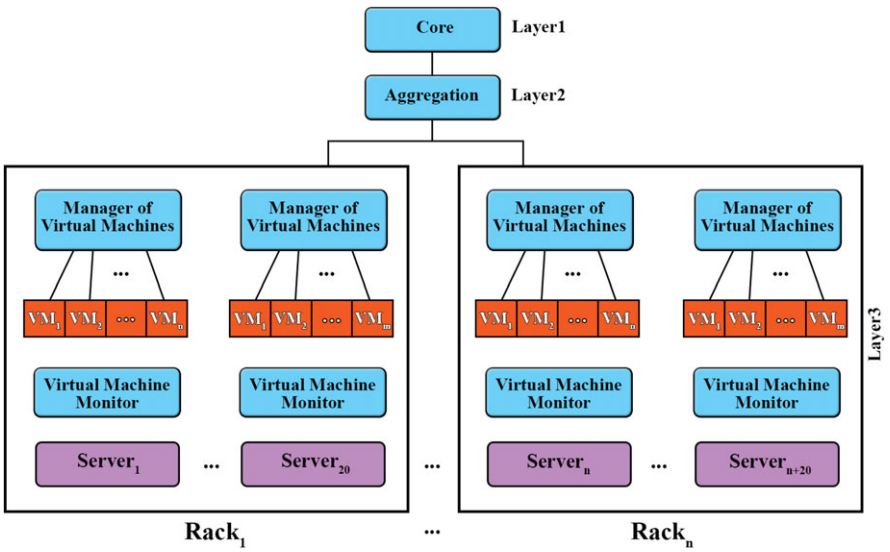

FIGURE 1 A layered view of a data center in a cloud environment

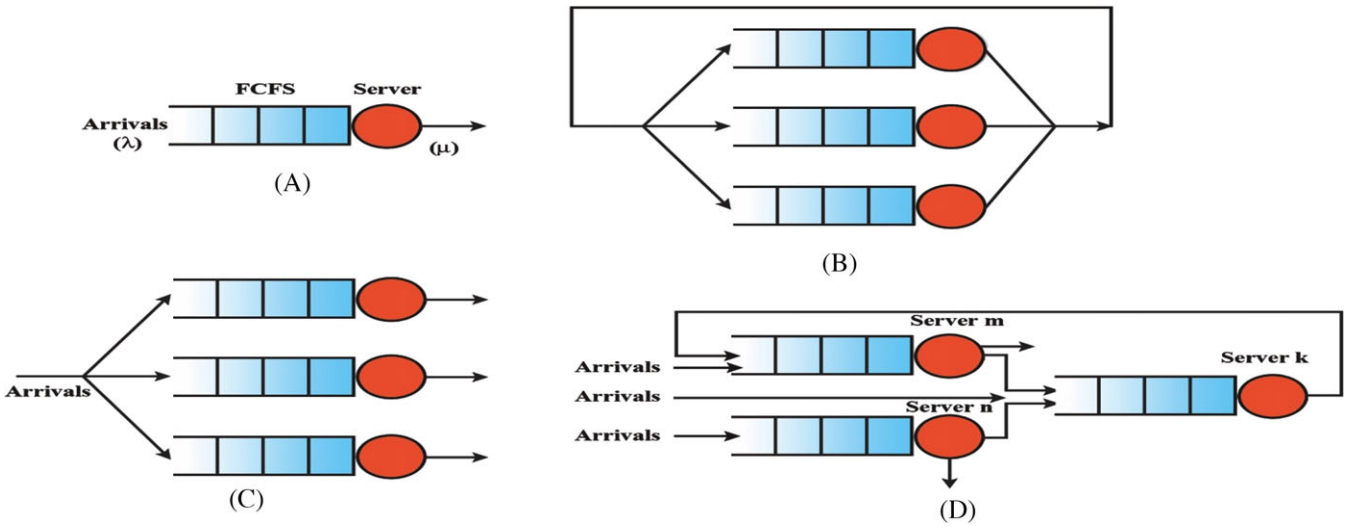

FIGURE 2 Some types of queuing models.[4] A, A simple queue; B, A closed-batch system; C, An open queuing system; D, A simple Jackson network

TABLE 1 Database sources for finding articles

Source	URL
IEEE Xplore	http://ieeexplore.ieee.org
ACM	http://dl.acm.org
Google Scholar	http://scholar.google.com
Springer	http://link.springer.com
Elsevier	http://www.sciencedirect.com

2.5 | Related regular surveys

Despite a comprehensive search of the literature in the sources mentioned in Table 1, we did not find any SMS or SLR for AQTMCC. However, there have been a handful of regular surveys in the field of AQTMCC. Here, we review them briefly because of the similarities between these works and our study.

Murugesan et al[25] presented one of the significant reports on various models of cloud computing using queuing theory for resource allocation. They found that the most examined model is M/M/c. The simplest model is M/M/1. However, their work lacks a discussion regarding the challenges in the modeling of cloud computing using queuing theory. Zhang et al[26] studied the state-of-the-art algorithms for resource provisioning in cloud computing. Techniques employed in these algorithms are categorized and analyzed systematically. However, their survey suffers from the lack of a detailed discussion and analysis of each of the algorithms.

Manvi and Shyam[27] provided a comprehensive review of the literature in the field of resource management on IaaS environments. The authors carefully defined the concept of resources and focused on some of the important resource management techniques, such as provisioning, allocation, mapping, and adaptation of resources. They classified the resources as either physical or logical. Performance metrics for resource adaptation schemes are also provided in their study. However, the authors put a little focus on auto-scaling.

Lorido-Botran et al[28] presented a review of auto-scaling techniques for elastic applications in cloud environments using queuing theory. The authors investigated many techniques proposed for automating application scaling. They classified these techniques into five categories: queuing theory, time series analysis, reinforcement learning, static threshold-base rules, and control theory. However, there is a gap for discussing the open issues and challenges in their review.

Santhi and Saravanan[29] surveyed queuing models for cloud computing. The queuing models, methods, parameters, and computations of each modeling technique are provided. However, their work lacks a discussion regarding the challenges in the modeling of cloud computing using queuing theory; in addition, a discussion of open issues and future topics that researchers should focus on is missing. Moreover, the limitations/weaknesses of modeling techniques are not mentioned in their survey.

3 | SEARCH METHOD

We conducted the guidelines provided in the works of Petersen et al[6] and Kitchenham et al[7] for SMS and SLR to provide better and highly reliable information about the topic of AQTMCC. Many process steps were performed in this study, as described in the following subsections.

3.1 | Research questions (RQs)

The aim of this SMS combined with SLR is to provide an overview of the current research and identify the gaps and future research directions on the topic of AQTMCC approaches in cloud computing. The overall objective is defined in the following seven RQs.

- RQ1: How many SLR, SMS, or regular reviews have there been since 2008 in the field of AQTMCC? Answering this question helps researchers to be familiar with the real history of AQTMCC.
- RQ2: What are the existing primary studies of AQTMCC, annual distribution, and their focus area? Answering this question enables us to classify the primary studies based on their investigated parameters and other metrics.
- RQ3: What are the publication statistics and venue of the existing primary studies on AQTMCC in the literature? Answering this question enables us to identify the distributions over popular publishers in this field.
- RQ4: What kinds of queuing models are used in the field of AQTMCC and which one is used frequently? By answering this question, we could provide a taxonomy of queuing models in cloud computing. Furthermore, it enables us to better draw the future of AQTMCC.
- RQ5: What experimental platforms have been used by the researchers for analysis and evaluation? By answering this question, we achieve compressive information about those platforms.
- RQ6: What queue disciplines were used in AQTMCC?
- RQ7: What are the open issues of AQTMCC research studies? Answering this question guides new researchers in determining the future path of AQTMCC.

3.2 | Database sources and search process

We organized our searches for this study in three phases. In all phases, some database sources have been used for searching research publications on AQTMCC; these database sources are shown in Table 1. In the Table, we show the name and the URL of the database sources. The study commenced in January 2017, and it was decided to search for publications in the period from January 2008 to January 2017.

Phase 1—finding SMS and SLR guidelines

In this phase, we made search strings to find guidelines for SMS and SLR. We considered the terms "guidelines for systematic mapping studies" and "guidelines for systematic literature review" as the main keywords with a set of related acronyms, namely, "SMS" and "SLR." We used the logical operators OR and/or AND to link the main keywords to their acronyms. Finally, after several tries, we found an appropriate search string: ("guidelines for SMS," "guidelines for systematic mapping study," "guidelines for SLR," or "guidelines for systematic literature review"). In the advanced search option of sources in Table 1, we applied the full search string by considering the title, abstract, and keywords. In some cases, we used some variation of the search string. A lot of articles were found, but we selected five of them for studying guidelines for SMS and SLR secondary studies.[6,7,13,14,30] Our selected studies on guidelines for writing SMS and SLR are listed in Table 2. The Table also gives the author names, titles, years, publishers, journals/conferences, and reference numbers.

Phase 2—finding related secondary studies

In this phase, we searched for SMS and SLR in the field of AQTMCC. We considered the following search strings: "queuing theory model for cloud computing," "systematic mapping study," and "systematic literature review." We tried the search string and its variations in the database sources of Table 1, but we did not find any SMS or SLR in the field of AQTMCC. Therefore, we made another search string for finding regular

TABLE 2 List of guideline studies used in our study

No.	Authors	Title	Year	Publisher	Journal/Conference
1	Petersen et al[6]	Guidelines for conducting systematic mapping studies in software engineering: an update	2015	Elsevier	Information and Software Technology
2	Kitchenham et al[7]	Using mapping studies as the basis for further research—a participant-observer case study	2011	Elsevier	Information and Software Technology
3	Kitchenham et al[30]	The value of mapping studies—a participant-observer case study	2010	Evaluation and Assessment in Software Engineering	Proceedings of Evaluation and Assessment in Software Engineering
4	Kitchenham et al[13]	The educational value of mapping studies of software engineering literature	2010	ACM	ICSE 2010 Education Theme
5	Petersen et al[14]	Systematic mapping studies in software engineering	2008	ACM	Proceedings of Evaluation and Assessment in Software Engineering

surveys, for example, "queuing theory model for cloud computing: a survey." Again, we tried the new search string in the database sources, and we found a handful of articles. We review them briefly in Section 2.5 for a comparison of their work and our work.

Phase 3—finding related primary studies

In the final phase of our search, based on the study topic and the proposed RQ, we defined the search keywords as the first step in formulating the search string. We considered the terms "queuing model," "queuing systems," "queuing theory," and "cloud computing" as the main keywords. We used the logical operators "OR" and "AND" to link the main keywords and have made some search strings, for example, "queuing model," "queuing systems," "queuing theory," and "cloud computing." We have searched for the strings, and we made some variations of them in the database sources of Table 1. For refining and filtering the hit list, we also used advanced search options of the database sources. In this way, we have investigated the keywords, titles, and abstracts of the articles.

3.3 | Study selection

In this subsection, our goal is to select the most relevant and important articles; inclusion and exclusion criteria were developed. Based on these criteria, the studies were selected by reading the title, abstract, and full text of the articles. Thus, we ensured that the results were related to the research area under study. As mentioned, we conducted our searches in three phases; in each phase, we considered the inclusion and exclusion criteria. The inclusion and exclusion criteria for each phase are shown in Table 3, and the adopted processes of article selection in the study are shown in Figure 3.

TABLE 3 Inclusion and exclusion criteria for article selection

Phase	Inclusion Criteria	Exclusion Criteria
1	• Article must describe SMS guidelines • Article must describe SLR guidelines	• Article which is in the form of books and technical reports • Article which is not written in the English language
2	• Article must report the queuing model for cloud computing • Article must describe the characteristics of cloud computing or a survey on AQTMCC • Article must address the challenges of modeling techniques for cloud computing	• Article which is in the form of books and nonreviewed technical reports • Article which is not written in the English language
3	• Article must have the modeling context in cloud computing • Article must describe the architecture of the data center and network • Article must aim for the modeling cloud computing either in general or in a specific aspect of cloud computing	• Papers which do not address the modeling of cloud computing • Gray literature and non-English papers • Non–peer-reviewed papers, keynotes, workshop reports, books, theses, and dissertations • Any obsolete or old version of a publication

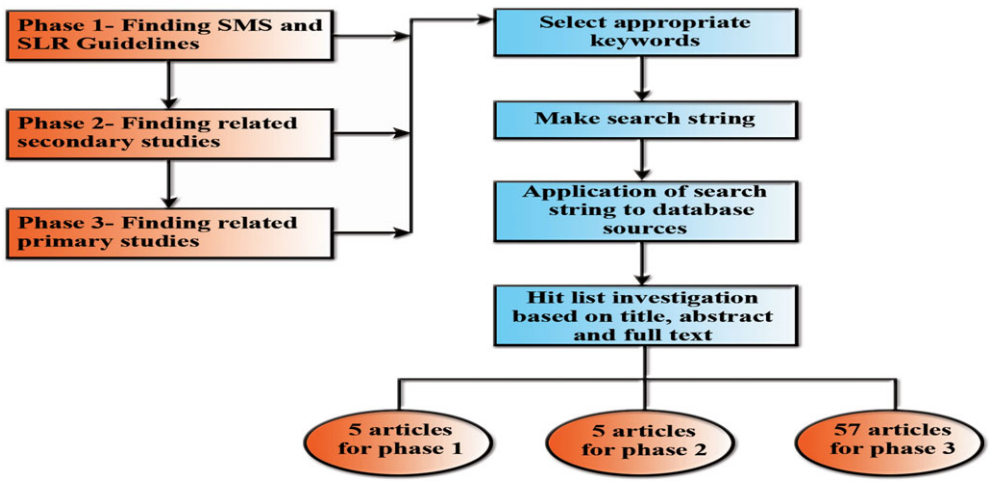

FIGURE 3 The adopted processes of article selection in the study

4 | PRIMARY STUDIES AND CLASSIFICATION SCHEMES

The primary studies selected in the SMS included 71 articles in the form of journal articles, conference proceedings, book chapters, and workshop proceedings since 2008. To analyze the primary studies of AQTMCC to answer our RQ, we classified the primary studies based on their focus area. As shown in Figure 4, our classification was based on the main aspects of cloud computing: performance, quality of service, workflow scheduling, energy savings, resource management, priority-based servicing, and reliability. Table 4 shows the classification of primary studies. In the Table, we see the category, description, and the number of studies in each category.

As mentioned before, we did not find any SMS or SLR in the field of AQTMCC, but we found some secondary studies in the form of a regular survey. The list of these secondary studies is shown in Table 5. In the Table, we see the author names, titles, years, publishers/journals, and reference numbers.

4.1 | Primary studies focusing on performance

In this subsection, we will investigate and analyze primary studies that focus on performance in the field of AQTMCC. As mentioned before, we found that 37% of primary articles under investigation focused on the performance aspect of cloud computing.

Ever[31] proposed a novel approximate analytical approach for the analysis of cloud computing centers with large numbers of servers. The author applied the $GI^x/M/S/N$ queuing model in the study; it is a multiserver queuing system with general inter-arrival time distribution, exponential service times, finite capacity, and batch arrivals. The author found that the presented analytical modeling approach worked for large state spaces

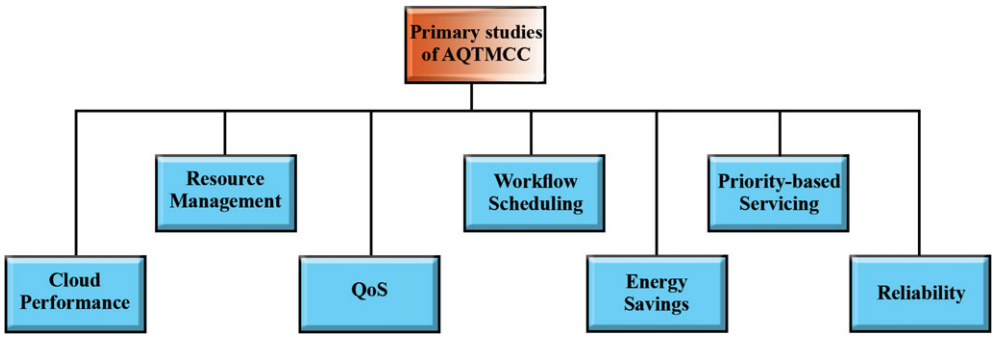

FIGURE 4 Primary-study classification based on focus areas

TABLE 4 A classification of primary studies

No.	Category	AQTMCC	Number of Studies
1	Performance	To optimize and evaluate the performance	26
2	QoS	To consider QoS	10
3	Workflow scheduling	For job optimizing scheduling, enhancing the scheduling, and providing job scheduling algorithms	9
4	Energy savings	To manage the energy, minimize the power consumption, and conserve cooling in data centers	9
5	Resource management	For resource sharing, resource provisioning, resource allocation, and resource management	11
6	Priority-based servicing	To prioritize the requests	3
7	Reliability	To study cloud service reliability and maximize the reliability	3
Total			71

TABLE 5 List of regular surveys on AQTMCC

No.	Authors	Title	Publisher/Conference	Year
1	Zhang et al[26]	Resource provision algorithms in cloud computing: a survey	JNCA (Elsevier)	2016
2	Santhi and Saravanan[29]	A survey on queueing models for cloud computing	IJPT	2016
3	Murugesan et al[25]	A status report on resource allocation in cloud computing using queuing theory	IJARCET	2014
4	Manvi and Shyam[27]	Resource management for Infrastructure as a Service (IaaS) in cloud computing: a survey	JNCA (Elsevier)	2014
5	Lorido-Botran[28]	A review of auto-scaling techniques for elastic applications in cloud environments	Journal of Grid Computing (Springer)	2014

with a high degree of accuracy. Mary and Saravanan[32] modeled the cloud center as [(M/G/1): (∞/GD)], a queuing system with an exponential arrival time, single-task arrivals, a task request buffer of infinite capacity, and a general discipline. Response time and waiting time distributions were obtained.

Chang et al[33] developed a novel approximate analytical model to evaluate the performance of active VMs in IaaS clouds using the M/G/m/m + K queuing system. The proposed model produced a more precise probability distribution of the number of jobs in the system, which can be used to obtain a set of performance metrics, including the mean queue length, the mean response time, and the blocking probability. Bai et al[34] investigated the heterogeneity of cloud data centers and the service process used in the servers of heterogeneous data centers, by constructing a complex queuing model. The complex queuing model is composed of two concatenated queuing systems: the main schedule queue (M/M/1/K queuing system) and the execution queue (M/M/c queuing system). The analyzed factors such as the traffic intensity or utilization of each execution server and the configuration of server clusters in a heterogeneous data center have a significant impact on the performance of the system.

Liu et al[35] presented a queuing model for the performance analysis of cloud services. Each server is then modeled as an M/G/1 queuing system. Their technique provided an accurate computation of important performance indicators, such as the distribution of waiting and the probability that a task will obtain immediate service. Varma et al[36] proposed a novel cloud computing model that is much useful for analyzing the cloud effectively to increase the performance indicators of cloud computing, such as mean queue length, utilization, throughput, and mean waiting time in the cloud system. They found that the dynamic allocation of resources could reduce mean delay and mean service time.

Khazaei et al[37] proposed an analytical model suitable for the performance evaluation of a cloud computing environment. They examined the effects of various parameters, such as arrival rate, the size of the task, virtualization degree on task rejection probability, and delay. Goswami et al[38] proposed a queuing model for studying the performance of computer services in cloud computing. Their model was useful for the prediction of service performance in cloud computing. Ait-Salaht and Castel-Taleb[22] generalized the model presented in the work of Goswami et al[38] by considering a multiserver queuing model with threshold queues and hysteresis in order to evaluate the performance of a data center. Several performance metrics, such as blocking probability, mean number of customers in the queue, and the number of departures, are evaluated according to different values of input parameters, such as queue size, the number of VMs or degree of virtualization, and utilization rate.

Raei et al[39] modeled and analyzed the performance of a cloudlet in mobile cloud computing (MCC). The work evaluated the effects of variations in a large set of parameters, such as workload (eg, request arrival rate), resource capacity (eg, a number of physical machines in a cloudlet or public cloud), and the connection failure rate on the request rejection probability and mean response delay. The authors used two M/M/1 queues and a number of M/M/C queues for their model. Vakilinia et al[40] proposed a performance model for systems with dynamic service demand, where the size of a job in terms of the number of tasks varied randomly during the remained time in the system. They obtained job blocking probabilities and distribution of the utilization of resources as a function of the traffic load under various scenarios for systems with both homogenous and heterogeneous VMs.

Khazaei et al[41] proposed a technique based on the Markov chain model for the performance evaluation of a cloud computing system. By this model, cloud providers can determine the performance metrics, such as the probability of immediate service, blocking probability, and the mean number of tasks in the system; also, they can determine the relationship between the queue size and the number of servers. Khojasteh et al[42] proposed a solution for the resource allocation of on-demand job requests in MCC to improve the performance of the system. The performance results indicated that the threshold-based priority scheme performed better and could be tuned to achieve the desired performance level.

Nguyen et al[43] proposed a novel three-state model, namely, ON, MIDDLE, and OFF, for cloud servers. The model is deployed in both single and multiple finite-capacity queues. Their approach reduces the waiting time for jobs and manages elastically the service capability for the system. Anupama and Keerthi[17] used a stochastic process to analyze the dynamic behavior of infinite servers over a single server. They studied the utilization factor, throughput, length of the queue, and waiting time of an infinite-server system. A good selection of a number of servers in such systems can increase the server utilization and throughput and reduce the queuesize.

Keller and Karl[44] investigated the optimal assignment of customers to the distributed resources with integrated queuing systems. The authors determined the response times depending on the number of used resources, which enables service providers to balance between resource costs and the corresponding service quality. They showed that adding more and more resources will, at one point, reduce the user-expected response time, only marginally. Therefore, the application provider can determine this point in advance and allocate resources accordingly. RahimiZadeh et al[45] proposed an analytical model based on the queuing network (M/G/1 model) to estimate the aggregated performance metrics of multitier applications in a data center. In addition, they introduced a methodology to measure the virtualized multitier applications (VMTA) characteristics, such as workload rate and demands on individual tiers, the transition probabilities of requests between tiers, and applications performance metrics. The results of their study demonstrated that the functionality and characteristics of tiers directly affect the overall performance of coexisting VMTAs.

Sun et al[46] proposed a reliability-performance correlation model that could analyze the performance of cloud service with fully considering reliability; the model is composed of two submodels, namely, reliability and performance. They used the Markov model for analyzing the two submodels. They illustrated the correlation between reliability and performance and the validation of the accuracy of the model by numerical examples. Liu et al[47] analyzed the performance of a cloud computing center considering resource sharing among VMs and its impact on service performance. The numerical example showed that the resource scheduling strategy of a cloud center has a great influence on performance. Moreover, they considered resource sharing among VMs.

Shi et al[48] formulated a multitier web system by queueing models. According to the characteristics of the system, servers in each tier can be modeled by M/M/n queueing, and the whole system forms a Jackson queueing network. A detailed methodology of the performance analysis was given. They tried to satisfy the performance requirements based on response time bounds. Khazaei et al[49] presented a performance model suitable for analyzing the service quality of large-sized IaaS clouds using interacting stochastic submodels. The validation of analytical results through extensive examples showed that cloud providers could obtain a reliable estimation of response time and blocking probability, resulting in the avoidance of service-level aggregation violation.

Akingbesote et al[50] modeled a typical cloud E-marketplace under a non-preemptive policy and evaluated the performance impact on consumer waiting time. They found that as the server utilization became high, the performance of their model based on waiting time was 80% better than the conventional nonpriority model. Their model is suitable for cloud providers where the cost model is prioritized based on the class of consumers. Pal and Pattnaik[20] proposed a queuing model with multiple servers and finite capacity to reduce the waiting time and queue length. Using the M/M/c and M/M/c/K queuing models, they provided a comparison study of waiting time.

Fakhrolmobasheri et al[51] proposed an analytical model based on stochastic activity networks for the performance evaluation of Infrastructure-as-a-Service (IaaS) cloud systems. In their work, they placed the input requests in the M/M/1 queueing model. In addition to performance, several real aspects of cloud systems were also considered, including failure/repair behavior of virtual machine monitors, virtual machines. Hanini and El Kafhali[52] provided a scheme based on the continuous Markov chain (CTMC) to manage virtual machine utilization in a physical machine with a workload control of the system. They analyzed the proposed scheme using mathematical evaluations of the QoS parameters of the system. The results demonstrated the usefulness of the proposed model to prevent overload in the system and to enhance its performances. El Kafhali and Salah[53] developed a queuing model to estimate the expected quality-of-service parameters, including response time, drop rate, throughput, and CPU utilization in cloud data centers (CDCs). In addition, they presented an energy consumption model to study and estimate the energy consumption in CDCs. The results showed that the proposed model is able to estimate the number of virtual machine instances required to achieve QoS targets under different workload conditions.

We thoroughly investigated and analyzed the primary studies in the field of AQTMCC that focused on performance. Our observations are summarized in Table 6. This analysis Table contains the names of publishers, publication year, author names, queuing models used in the paper, queue disciplines, advantages, disadvantages, experimental platforms, journals/conferences publishing the paper, and the parameters and computations performed in the study.

4.2 | Primary studies focusing on QoS

Of the studied primary articles, 14% focus on the QoS of AQTMCC. In this subsection, we investigate and analyze them.

Vilaplana et al[5] proposed a queuing model to study the QoS of computing service in cloud computing, where the cloud architecture was modeled using well-known open Jackson networks of *M/M/m* servers. They concluded that the model could be very useful for tuning service performance, for example, response time, thus guaranteeing the service-level agreement (SLA) between the client and the service provider. Murugan et al[54] proposed the M/M/C: ∞/∞ queuing model. The authors focused on the quality-of-service improvement in a cloud server. The authors used Mozilla's Firebug and Firefox to analyze the response time.

Kirsal et al[55] investigated an approximate Markov reward model approach to obtain QoS measures. They applied an M/M/C queueing model with infinite capacity of the queue; also, they considered finite buffer capacity. The evaluation results revealed that data center failures and repairs affected the QoS of the system significantly. Therefore, system availability is important for cloud system designing and modeling.

Xiong and Perros[56] obtained the response time distribution of a cloud system modeled on a classic M/M/m open network, assuming that the inter-arrival and service times had an exponential distribution. For a given service resource, the authors obtained the level of QoS services that could be guaranteed in terms of response time. Khomonenko et al[57] studied a class of M/M/C/n systems with cooling. Cooling can be understood as decrypting, emptying cache, logging, etc. All these factors influence the cloud performance, thereby affecting QoS. Considering the non-Markov cooling process, the studied model was a multichannel model, featuring an unbounded queue; it determines the practical importance of the presented results in assessing the efficiency of cloud systems with multiple processing nodes.

Rajendran and Swamynathan[58] proposed the M/M/c broker queuing model to reduce the waiting time, thus improving the response time to the customer; it is important to reduce the waiting time to improve the QoS in service discovery. They concluded that to have an efficient system in handling service discovery requests in cloud computing, it is necessary to have a multibroker system.

Vilaplana et al[59] presented a new application of the cloud computing paradigm by designing a system model applied to e-Health. They used a combination of two M/M/m systems in a sequence. Their work revealed that to provide good QoS, in terms of mean waiting times, the waiting time between the first and the second phase tends to stabilize. Vilaplana et al[60] presented a model for designing cloud computing architectures, which has QoS. They presented two main alternatives for doing so. The first one was based on queuing theory and open Jackson networks, and the second one was to develop new event-driven simulation policies. The simulation results demonstrated the usefulness of the models with guaranteeing the QoS under ideal conditions and when scaling the system.

Cho and Ko[61] investigated the stabilization of the mean virtual response time in a single-server processor-sharing (PS) queueing system with a time-varying arrival rate and a service rate control (GIt/GIt/1/PS queue). They proposed a mechanism to stabilize the mean virtual response time

TABLE 6 An overview of existing primary studies focusing on performance

Publisher	Year	Authors	Queuing Models (Discipline)	Advantages	Disadvantages	Experimental Platforms	Parameters/ Computations
MDPI	2018	Fakhrolmobasheri et al[51]	M/M/1 (FCFS)	• Heterogeneous physical machines and virtual machines	• No real-world experiments • Not scalable	• Solving analytical model by Möbius tool	• Performance • Availability • Power consumption
Springer	2018	El Kafhali and Salah[53]	M/M/K/ m (FCFS)	Estimating correctly the number of needed VMs	• Not considering cloud-hosted containerized services	• JMT tool • SHARPE	• Energy consumption • CPU utilization • QoS parameters
Elsevier	2017	Raei et al[39]	M/M/1 M/M/C (FCFS)	Finding bottlenecks and best settings of parameters	• The cost of provisioning in the public cloud is ignored • No implementing on a real-world cloud	• Using SHARPE software package	• Workload • Connection failure • Request rejection probability • Effects of queue size on request rejection probability • Mean response delay
ACM	2017	Hanini and El Kafhali[52]	M/M/c (FCFS)	Maintaining QoS at an acceptable level	• Not considering SLA constraints	• MATLAB software	• Response time • Loss probability • Mean no. of requests
Springer	2016	Ever[31]	GI^x/M/S/N (FCFS)	• Low computations • High accuracy • Large number of servers • Flexibility	• Not considering availability issues	A simulation program is written in C++	• Availability • Server failure • Number of servers • Arrival rate
IEEE	2016	Chang et al[33]	M/G/M/m + K (FCFS)	• High accuracy • Can be applied to any service time distribution	• Single task of jobs • No batch arrivals	• Using Maple 7 • Simulations in Software Package Arena	• Mean response time • Blocking probability • Mean number of jobs • Probability of immediate service
IEEE	2016	Khazaei et al[41]	M/G/m/ m + r (FCFS)	• Discussing heterogeneous systems	• A monolithic analytical model that is restrictive in terms of simplicity and computational cost	• Using Maple • Using simulation engine Artifex for validating the analytical solution	• Relationship between the number of servers and input buffer size • Mean number of tasks • Blocking probability • Probability of immediate service • Response time
IEEE	2016	Khojasteh et al[42]	M/M/L/L (FCFS)	• Flexible resource allocation performance • Forked tasks area given full priority over newly arriving tasks	• Setting limitations for forked tasks • Assuming new arriving job demands service for a single task • No simulation	• Using Maple • The model solved in a number of different scenarios	• Mean task delay • Task blocking probability • Mean task service time • Successful provisioning probability • Total rejection probability
Springer	2016	Nguyen et al[43]	M/M/C (FCFS)	• Providing an elastic architecture • Reducing service waiting time	• Complexity of the system due to multiple-queue model	• CloudSim simulation tool	• Service waiting time

(Continues)

TABLE 6 Continued

IEEE	2016	Sun et al[46]	M/M/C (FCFS)	• Performance and reliability • Considering physical machine and VM failures in heterogeneous environment • Multitask jobs	• No simulation or implementation	• Numerical analysis	• Effects of reliability on performance • Efficient service rate • Physical machine failures • VM failures • Service rate of cloud service
IEEE	2016	Shi et al[48]	• M/M/n for servers in each tier • Jackson queuing network (FCFS)	• Single-server and multiserver models • Optimizing soft resource allocation	• Assuming that the threads are homogeneous with the same service rate	• Using RUBiS system as a testing cluster in experimental study	• Server utilization • Mean service and response time • Mean number of requests • Total delay • Waiting time • Service time
ProQuest	2016	Pal and Pattnaik[20]	M/M/C, M/M/C/K (FCFS)	• Reducing waiting time • Reducing queue length • Comparing waiting time of queue models	• Not considering heterogeneity of servers • No implementation in real-cloud environment	• Using CloudSim simulation tool	• Arrival rate • Service rate • Server utilization • Average number of customers in system • Waiting time
Hindawi	2015	Bai et al[34]	M/M/1/K M/M/C (FCFS)	• Considering heterogeneity of modern data centers • Performance of heterogeneous data centers	• Complexity of the system	• CloudSim simulation tool	• Mean response time • Mean waiting time • Traffic intensity or utilization on each execution server • Configuration of server clusters
Springer	2015	Liu et al[35]	M/G/1 (FCFS)	• Dividing service request into subtasks • Considering heterogeneity • Computing the effect of scheduling on performance	• It does not construct an accurate performance model • No simulation or implementation	• An illustrative example presented	• Mean response time • Blocking probability • Resource sharing among VMs • Probability of immediate service • Waiting and completion times of requests
IEEE	2015	Ait-Salaht and Castel-Taleb[22]	M/M/S with threshold queues and hysteresis (FCFS)	• VMs activated or deactivated according to the intensity of user demand • Flexibility of different models	• Considering homogeneous servers • No simulation or implementation	• Presenting some numeric examples	• Degree of virtualization • Blocking probability • Mean number of customers in buffer • Mean number of departures • Number of VMs • Utilization rate

(Continues)

TABLE 6 Continued

Elsevier	2015	Vakilinia et al[40]	M/M/∞ (FCFS)	• Homogeneous and heterogeneous VMs • Multitask jobs • Different classes of tasks	• Dynamic assignment of VMs is time consuming	• Numerical analysis • Simulation for verifying the numerical results	• Joint probability distribution of the number of jobs • Job blocking probabilities and distribution of the utilization resources
Elsevier	2015	RahimiZadeh et al[45]	M/G/1 (FCFS)	• Determining the model flexibility and complexity • Can be extended to the M/G/C model	• The resource utilization must be less than one • In case of system saturation, this model would not be valid	• Evaluation of RUBiS Virtual Appliance and Media Wiki as two multitier applications is virtualized mode configured in Xen environment	• The transaction probability among tiers • Response time in each tier • Caching probability for each tier • CPU utilization • Disk utilization • Maximum deviation between the model and experimental results
IJCSET	2014	Anupama and Keerthi[17]	M/M/1, M/M/∞ (FCFS)	• Stochastic description of M/M/1 and M/M/∞ models • Comparing the two models M/M/1 and M/M/∞	• No simulation or implementation • No mention of future works	• Numerical analysis	• Arrival rate
ACM	2014	Keller and Karl[44]	M/M/1 (FCFS)	• Modeling the system with and without a queuing system • Discusses the response time reduction	• Requests are assigned to sites that provide a single computer resource (eg, a single fast server)	• Numerical evaluation	• Response time • Queuing delay • Effects of queuing delay on response time
Springer	2014	Liu et al[47]	M/M/1 (FCFS)	• Resource sharing among VMs • Considering failures • Heterogeneity of servers	• Not considering the buffer size of the subtask • No simulation or implementation	• Numerical analysis	• Arrival rate • Service rate • VM failures • Physical machine failures • Network failures • Waiting time • Completion time • Service time
IEEE	2014	Akingbesote et al[50]	M/M/C for each service station (FCFS in each class)	• Reduces delay • Considering heterogeneity of servers • Priority-based servicing	• Not considering the total cost of service • Increasing waiting time for low-priority jobs	• Using Arena Discrete Event Simulator version 14	• Arrival rate • Service time • Waiting time • The probability that the system is busy • Server utilization
ProQuest	2013	Mary and Saravanan[32]	M/G/1: (∞/GD) (General Discipline)	• Simplicity of the models	• Not analyzing the results using simulation	–	• Service time • Mean number and standard deviation of tasks • Blocking probability • Immediate service probability

(Continues)

TABLE 6 Continued

IEEE	2013	Khazaei et al[49]	M/M/1 (FCFS)	• Insights for capacity planning and delay controlling	• Assuming error-free conditions • Assuming a fixed number of physical machines in each pool	• Using Maple 15 from Maple Soft, Inc • Numerical analysis	• Arrival rate • Service time • Task rejection • Virtualization degree • Reliable response time • Waiting time • Resource utilization
IEEE	2012	Varma et al[36]	M/M/1 (FCFS)	• Reducing congestion in buffers • Reducing mean delays • Reducing queue length	• Not considering heterogeneity	• Numerical analysis	• Arrival rate • Service rate • Mean service time • Mean delay • Mean queue length • Utilization • Throughput
IEEE	2012	Goswami et al[38]	M/M/S (FCFS)	• Dynamically create and remove VMs • No VM migration	• No simulation or implementation in real-world cloud	• Using MATLAB for validation of analytical results	• Arrival rate • Service rate • Service performance predictions
IEEE	2011	Khazaei et al[37]	M/G/m (FCFS)	• Large number of servers • General service time • Flexibility	• Not considering burst arrivals • Not considering subtasks	• Simulation engine Artifex by RSoftDesign	• Request response time • Number of tasks in the system • General service time

of a GIt/GIt/1/PS queue. Melikov et al[62] explored queuing management with feedback in cloud computing centers with large numbers of web servers to analyze the QoS metric of cloud computing. They found that the result of their study is applicable in the real-cloud system in order to calculate the QoS metrics depending on the application area.

The primary studies in the field of AQTMCC that focused on QoS were thoroughly investigated and analyzed. Our observations are summarized in Table 7. This analysis Table contains the names of the publishers, publication year, author names, queuing models used in the study, queue disciplines, advantages, disadvantages, experimental platforms, journals/conferences publishing the paper, and the parameters and computations performed in the study.

4.3 | Primary studies focusing on workflow scheduling

In this subsection, we investigate and analyze primary articles that focus on workflow scheduling in the field of AQTMCC. As mentioned before, 13% of primary articles that we studied focused on the workflow aspect of cloud computing.

He et al[63] proposed a novel algorithm called QTJS to optimize VM allocation and job scheduling. The authors modeled the computing process of each VM using an M/M/1 queuing model. To simulate heterogeneous environments, they selected four different types of the host to build a heterogeneous Hadoop cluster. Extensive experiments showed that their QTJS algorithm reduced job execution time, and it outperformed the efficiency of the other three compared algorithms.

Eisa et al[64] proposed a model for cloud computing scheduling based on multiple queuing models. The model consisted of four modules: multiple waiting queues for incoming requests, a global scheduler based on the scheduling algorithm, local schedulers, and waiting queues for each local scheduler. Experimental results indicated that the model increases the utilization of a global scheduler and reduces the waiting time. Li[65] built a non-preemptive priority M/G/1 queuing model for the jobs by the analysis of the differentiated QoS requirements of the cloud computing resources of the users' jobs. Then, the author gave the corresponding strategy and algorithm to get the approximate optimistic value of service for each job in the corresponding non-preemptive priority M/G/1 queuing model.

Dutta et al[66] proposed a job scheduling algorithm for efficient cloud computing resource management using the M/G/1 queuing model with non-preemptive priority. Moreover, they proposed scheduling heuristics that could be incorporated at a data center level for selecting an ideal host for VM creation. They assumed that the users' jobs had different classes with different priorities and classified them into several classes. Rashidi and Sharifian[67] proposed a novel algorithm for task assignment in mobile cloud computing systems in order to reduce offload duration time while balancing the cloudlets' loads. They applied an M/M/∞ queuing model for public cloud and an M/M/Vs/∞ queuing model for each cloudlet. The simulation results indicated that the proposed algorithm decreased the completion time, the mobile users' average battery-power consumption, and the rejection rate of offloaded tasks in the system. At the same time, it balanced the load of the cloudlets so that they could all be fully utilized.

Peng et al[68] introduced a fine-grained cloud computing system model with an optimization task-scheduling scheme. They applied an M/M/1 model and a series of M/M/1/m queuing models. Srivastava[69] proposed a job scheduling algorithm to improve the overall QoS of the e-business architecture being implemented using a cloud computing environment. The scheduling algorithm is based on a GI/G/3/n/k queuing model. Sundararaj[70] proposed a queue-based efficient algorithm, referred to as QAnt-Bee, for the optimal assignment of tasks in a mobile cloud computing environment. The proposed algorithm minimized the average completion time of the tasks, power consumption, and the offloaded

TABLE 7 An overview of existing primary studies focusing on quality of service

Year	Publisher	Authors	Queuing Models (Discipline)	Advantages	Disadvantages	Experimental Platforms	Parameters/ Computations
2018	Cornell University	Cho and Ko[61]	GIt/GIt/1 /PS	• Ease of use • Simplicity	• Not considering time-varying queues	• Not mentioned	• Response time
2018	Springer	Melikov et al[62]	M/M/c	• High accuracy • Low computation time	• Not considering heterogeneity	• Numeric analysis	• Server repair time • Blocking probability • Wait and response times
2016	IEEE	Khomonenko et al[57]	• M/M/ C/n/R • A/B/C/n (FCFS)	• Examining a multichannel non-Markovian queue is realistic • Using local content to speed up the whole business process	• Considering the system context is time consuming • Managing the context is expensive	• A Java program has been written to implement the described numeric method	• Arrival rate • Service rate • Waiting time • Sojourn time • Queue length • Number of requests in the system
2016	Springer	Rajendran and Swamynathan[58]	M/M/C (FCFS)	• Using cloud brokers • Comparing several queuing models for finding minimum waiting time	• No evaluation in a real-cloud environment • Lack of scalability	• Numerical analysis and mathematical formulation	• Arrival and service rates • Response and waiting times • Utilization factor • Queue length
2015	ERP (Enhanced Research Publications)	Murugan et al[54]	M/M/C: ∞/∞ (FCFS)	• Defining several scenarios and interpreting the results	• Considering homogeneous servers is not realistic in a cloud system	• Using the Microsoft Windows Azure platform	• Service time • Response time • Traffic intensity • Number of servers • DLE server • Delay in queue • Queue length
2015	IEEE/ACM	Kirsal et al[55]	M/M/C (FCFS)	• Considering the effects of failures and repairs on QoS • Considering availability • Queue capacity is infinite scalable	• Not considering the costs of failures and repairs	• Numerical analysis using examples	• Arrival rate • Service rate • Failure and recovery times • Number of busy servers • Stability of the system • Mean queue length
2015	Western Sydney	Vilaplana et al[60]	M/M/1, M/M/C (FCFS)	• Considering energy consumption • Considering parallel processing • Improving QoS	• Not considering complicated jobs • Model is not realistic • Not implemented in real clouds	• Using CloudSim 3.0.2 software	• Arrival rate • Service rate • Execution time of jobs • Energy consumption • Resource utilization

(Continues)

TABLE 7 Continued

2014	Springer	Vilaplana et al[5]	M/M/1, M/M/m (FCFS)	• Reducing response time • Finding bottleneck of the system	• Not considering user variability and reliability of the cloud platform • Considering that only one database is not realistic	• Using Open Stack • The model is implemented using Sage 5.3 mathematical software	• Arrival rate • Client bandwidth • Service rate • Response time of the servers • Mean size of the files that are sent to clients via the Internet
2013	BMC (BioMed Central)	Vilaplana et al[59]	M/M/C (FCFS)	• Adaptability • Reducing waiting time to improve QoS • Saving resources • Scalability	• Lack of accurate model validation • Leaving aside reliability, availability, and security	• Using the queue simulator server Queue 2.0	• Arrival rate • Service rate • Response time • Mean access rate to database • Waiting time
2009	IEEE	Xiong and Perros[56]	M/M/m (FCFS)	• Finding the relationships among the maximal number of customers, the minimal service resources, and the highest level of services	• Not scalable • The model has limited application and does not fit most practical systems	• Numerical analysis	• Arrival rate • Service rate • Percentile delay • Probability and cumulative distributions of response time • Waiting time

tasks' rejected rate with the support of a queue model. Moreover, the ''cloudlets'' load is balanced. Narman et al[71] proposed Homogeneous Dynamic Dedicated Server Scheduling (DDSS) and Heterogeneous Dynamic Dedicated Server Scheduling (HDDSS) and explained the scheduling procedure. Then, they derived the upper- and lower-bound performance metrics, average occupancy, drop rate, average delay, and throughput, for each class of application in the proposed scheduling algorithm by using queuing theory.

We thoroughly investigated and analyzed the primary studies in the field of AQTMCC that focused on workflow scheduling. Our observations are summarized in Table 8. This analysis Table contains the names of the publishers, publication year, author names, the queuing models used in the paper, queue disciplines, advantages, disadvantages, experimental platforms, journals/conferences publishing the paper, and the parameters and computations performed in the study.

4.4 | Primary studies focusing on energy savings

In this subsection, we investigated and analyzed the primary articles that focus on energy savings in the field of AQTMCC. Of the studied articles, 13% fall in this category.

Liao et al[72] proposed a mathematical model that used queuing theory to determine the activation thresholds of servers to guarantee performance requirements in terms of waiting time to minimize power consumption through the dynamic management policies that switch on/off a certain group of servers. They developed a mathematical model using the M/M/n + m1 + m1 queuing model to determine the activation thresholds of the servers. Bi et al[73] proposed a temporal request scheduling algorithm (TRS) that considered temporal diversity. The authors compared TRS with some existing scheduling methods and found that TRS achieves a higher throughput and lower grid energy cost for a green cloud data center while meeting each request's delay requirement. In their work, the authors used the M/M/1/N/∞ queuing model.

Akbari et al[74] applied a weighted linear prediction technique and M/M/1 queuing theory for enhancing the energy efficiency of cloud data centers. They simulated the effect of various workloads on the energy consumption of the cloud system using CloudSim or similar software. Cordeschi et al[75] developed an optimal minimum-energy scheduler for the adaptive joint allocation of task sizes, computing rates, communication rates, and communication powers in virtualized networked data centers. By analysis, the authors showed that the amount of data travels through the communication link and maximum bandwidth of the link were highly influential on the network performance and consumed energy in the channel.

Cheng et al[76] provided a power-saving job scheduling protocol based on a vacation queuing model for a cloud computing system. The authors suggested a task scheduling protocol based on similar jobs for reducing power consumption. Simulations proved that the suggested protocol decreased the power usage of cloud computing systems efficiently while fulfilling task performance. Shi et al[77] developed an efficient energy-saving method to reduce the huge energy consumption in cloud data centers. They improved the M/M/1 queuing theory–predicting method with a better

response time during the rapidly growing period and reduced the reserved resource in a steady sequence. When investing for on-site renewable energy generation is costly, data center operators can also sign up with a local renewable energy generator. Ghamkhari and Mohsenian-Rad[78] proposed an analytical model to determine the profits in a data center with renewable power generation and included service-level agreements in their model. The authors applied the M/M/m queuing model in their work.

Balde et al,[79] by taking into account a fat-tree topology, proposed a new analytic model with different activation thresholds in order to reduce power consumption in data centers. They used the queuing theory based on a mathematical model. They found that their proposed model outperforms previous energy-aware queuing theoretical models. They modeled the system as an $M/M/k^{3/4}$ queue. Chunxia and Shunfu[80]

TABLE 8 An overview of existing primary studies focusing on workflow scheduling

Year	Publisher	Authors	Queuing Models (Discipline)	Advantages	Disadvantages	Experimental Platforms	Parameters/ Computations
2018	Springer	• Sundaraj[70]	M/M/∞	• Trying some queue-based algorithms, • The "cloudlets" load is balanced	• No effective online scheduling algorithm	• Not mentioned	• Drop rate • Cloudlets • Load task • Completion time
2017	Springer	• Narman et al[71]	M/M/1/N M/Mi/m/N	• Dynamic scheduling algorithm • Considering homogeneous and heterogeneous servers	• Not considering the effects of the heterogeneity level of servers and on performance	• Not mentioned	• Drop rate • Throughput • Performance • Server utilization
2017	Elsevier	Rashidi and Sharifian[67]	• M/M/∞ for public cloud M/M/ Vs/∞ for each cloud set (FCFS)	• Decreasing completion time • Less power consumption • Balancing cloudlets load • Increasing utilization	• It is assumed that the service time of all servers is equal • Considering homogeneous servers	• Simulation in MATLAB	• Completion, wait, service, response times • Number of tasks sent to the public cloud • System cost • Stable probability of each M/M/C • Probability of entering a new task in the cloudlets queue
2016	Springer	He et al[63]	M/M/1 (FCFS)	• MapReduce for big data analysis • Less job execution and waiting time	• Modeling M/M/1 is not realistic for a cloud environment	• Deploying the algorithm in a computer cluster with nine nodes	• Throughput • Job execution time and costs • VM workload efficiency • Job waiting time • Network and CPU workload • Task delay time • Resource utilization
2015	Springer	Peng et al [68]	M/M/1, M/M/1/m (FCFS)	• Scalability • Reduces response time • Efficient task scheduling	• Not considering VM failures, VM migration, and burst arrivals	• Using MATLAB R 2012a by Math Works, Inc	• Arrival rate • Service rate • Response time • Mean waiting time
2014	ProQuest	Eisa et al[64]	M/M/1, M/M/S (FCFS)	• Increases utilization • Reduces waiting time • Realistic modeling of a cloud	• Considering homogeneous servers	• Using Maple	• Arrival rate • Service rate • Queue length • Residence time • Utilization • Throughput

(Continues)

TABLE 8 (Continues)

2012	IEEE	Dutta et al[66]	M/G/1 (in each class with the same priority, jobs are processed in FIFO)	• Classifying the job priorities into several classes	• No realistic assumption for cloud • The model has not a general, closed-form distribution	• Implementing the advanced job scheduling algorithm • Numerical analysis	• Average time spent in queue • Throughput • Probability that server is empty • Queue length • Waiting time • Cost model
2012	CiteSeer	Srivastava[69]	GI/G/3/n/K, M/M/1 (FCFS)	• Assurance of QoS • Comparing the queuing models M/M/1 and GI/G/3/n/K • The model is effective and efficient	• Not considering cost of service • Considering homogeneous servers	• A software developed in Java 2.0	• Arrival rate • Service time • Response time • Queue length • Total number of requests
2009	IEEE	Li[65]	M/G/1 (In each class with same priority, jobs are processed in FIFO)	• Considering jobs with different classes and priorities • Maximizing profit for cloud providers • QoS	• Not computing the number of jobs for a given QoS • It is not clear how to regulate the service rate	• Numerical analysis	• Arrival rate • Service time • Traffic intensity of jobs • Waiting time • Mean number of jobs in each class • Total time a job spent in cloud • Cost function for jobs

proposed an energy-saving strategy based on the multiserver vacation queuing theory that switches servers between "on" and "sleep" in groups. They modeled the data center as an M/M/c vacation queuing system. They found that a smaller group is superior from the perspective of energy savings and that a larger group is superior from the perspective of response delay.

We thoroughly investigated and analyzed the primary studies in the field of AQTMCC that focused on energy savings. Our observations are summarized in Table 9. This analysis Table contains the names of the publishers, publication year, author names, the queuing models used in the study, queue disciplines, advantages, disadvantages, experimental platforms, journals/conferences publishing the paper, and the parameters and computations performed in the paper.

4.5 | Primary studies focusing on resource management

In this subsection, we investigated and analyzed the primary articles that focus on resource management in the field of AQTMCC. Of the studied articles, 15% fall in this category.

Shi et al[81] proposed a novel resource provisioning method, including VM provisioning for hosting service and VM placement in servers. By using the M/M/c queuing model, the proposed method determines how many VMs should be provided for each service. Experimental results showed that the proposed method achieved a better performance than the baseline methods. The work reported by Ellens et al[82] concerned the assignment of reserved and shared resources in a data center to a number of customers. They used an M/M/C/C queueing model with multiple service classes to determine the blocking probability of customer requests. The model is also used for data center dimensioning purposes, for example, determining the size of reserved and shared resource pools.

Xiong and Perros[83] proposed an approach for a resource allocation problem in a typical service provider's cluster computing environment, whereby minimizing the total cost of computational servers to a customer. They modeled the system using M/M/1 and M/M/1/B queuing systems. The authors considered QoS metrics, including percentile response time, cluster utilization, packet loss rate, and cluster availability. Xiong and He[84] investigated the problem of resource allocation for power management in MapReduce clusters using an M/M/c queuing model. In their numerical experiments, they found that the proposed approaches were applicable and efficient in solving resource allocation problems for power management in MapReduce clusters.

Casalicchio and Silvestri[85] proposed mechanisms for SLA provisioning in cloud computing. They used an M/M/m queuing model to determine the minimum number of VMs needed to handle a given load and to satisfy the service-level objective. Hu et al[86] proposed a heuristic algorithm to determine the job scheduling policy and server allocation strategy to minimize the number of servers needed for service. The authors applied the M/M/c queuing model for resource provisioning. Furthermore, they presented an algorithm to determine the minimum number of required servers. The proposed scheduling disciplines were evaluated analytically.

Nan et al[87] proposed a cost-effective resource allocation optimization approach for a multimedia cloud based on a queuing network analysis. In the study, the authors used M/M/1 and M/G/1 queuing models to capture the relationship between the service response time and the allocated resources. Nan et al[88] proposed optimal resource allocation policies for a multimedia cloud in both the single-class service case and the multiple-class service case based on M/M/1, M/M//1, and $M/H_m/1$ queuing models, where H_m represents the hyper-exponential distribution. In each scenario, they formulated and solved two problems: the response time minimization problem and the resource cost minimization problem. Nan et al[89] studied the resource allocation problem in priority-based servicing with two objectives: (1) minimizing the resource cost and (2) minimizing the response time for cloud service providers. In their study, they modeled the cloud system by an M/M/s queuing model.

TABLE 9 An overview of existing primary studies focusing on energy savings

Year	Publisher	Authors	Queuing Models (Discipline)	Advantages	Disadvantages	Experimental Platforms	Parameters/ Computations
2018	Elsevier	Balde et al[79]	$M/M/k^{3/4}$	• Saves more energy • Considering batch arrival	• Not considering heterogeneity	• Numerical analysis and simulation	• Waiting time • Number of jobs in a system • Power consumption
2018	Springer	Chunxia and Shunfu[80]	M/M/c (FCFS)	• Energy reduction • Reduces response delay	• Not considering heterogeneity	• Simulation in MATLAB	• Number of busy servers • Mean sojourn time
2016	Elsevier	Bi et al[73]	M/M/1//N/∞ (FCFS)	• Increasing throughput • Reducing grid energy cost • Meeting delay requirements	• Not implementing in a real cloud • Considering only one type of requests	• Experimented with real-life requests and real-life grid price	• Loss probability • Number of requests each active server can execute • Energy consumption
2016	IEEE	Akbari et al[74]	M/M/1 (FCFS)	• Improving energy consumption • Predicting throughput of applications • Reducing violation rates	• No implementation in a real-cloud environment • The model is not realistic	• Using the CloudSim simulation tool	• Arrival rate • Service rate • Energy consumption • Response time • Resource utilization
2015	IEEE	Liao et al[72]	M/M/n+ m1 + m2, where n, m1, and m2 are servers in their groups (FCFS)	• Minimizing power consumption • Guaranteeing performance requirements • Providing dynamic policies	• Complex computations • No implementation in a real-cloud environment • No considering heterogeneous data centers	• Using an analytical approach	• Arrival rate • Service rate • Waiting time • Energy consumption • Determining activation thresholds of servers
2015	IEEE	Cheng et al[76]	M/G/1 (FCFS)	• Considering heterogeneous compute node • Reducing power usage • Meeting task performance	• No automatic energy-saving management and performance optimizations	• Setting up a dynamic environment using a discrete-event simulation tool in MATLAB	• Arrival rate • Service rate • Waiting time • The idle, sleep, running, and recovering times • Task sojourn time
2014	Springer	Cordeschi et al[75]	M/G/1 (FCFS)	• Reducing energy consumption • Average energy loss is less • Considering granularity of jobs	• No emphasis on internal switching costs in VMs • No implementation in a real-cloud environment	• Numerical evaluation of the solution in MATLAB	• Arrival rate • Service rate • Energy consumption • Maximum tolerated processing delay

(Continues)

TABLE 9 (Continues)

2013	IEEE	Ghamkhari and Mohsenian-Rad[78]	M/M/m (FCFS)	• Reducing cost of electricity • Maximizing profit for data centers • Considering QoS and SLA	• Considering fixed price for all requests • Not considering resource rental cost	• Analytical analysis • Simulation using an event-based simulator	• Incoming workload • Price of electricity • Maximum waiting time • Service rate • Service money of data center
2011	IEEE	Shi et al[77]	M/M/1 along with the linear prediction method (LPM) (FCFS)	• Reducing energy consumption • Using log for resource reservation • QoS • Renewable method	• Not tested in a real cloud • Does not consider diverse workload • Assuming the same arrival rate in a short period	• Using the CloudSim simulation tool	• Arrival rate • Service rate • Resource utilization • Average number of customers • Energy consumption • Violation rate

Song et al[90] proposed a queuing-based approach for task management and a heuristic algorithm for resource management. By simulation, they found that the proposed solution provided cost-effective and flexible task and resource management than the state-of-the-art approaches. Vakilinia and Cheriet[91] modeled the data stream as a two-dimensional semi–birth-death queuing process to calculate the required resources and active servers for serving the data stream. They found that the preemptive resource allocation is too sensitive to the dynamic nature of the system and that the static primary jobs provide a better test bed for the secondary processing jobs.

We thoroughly investigated and analyzed the primary studies in the field of AQTMCC that focused on resource management. Our observations are summarized in Table 10. This analysis Table contains the names of the publishers, publication year, author names, the queuing models used in the study, queue disciplines, advantages, disadvantages, experimental platforms, journals/conferences publishing the paper, and the parameters and computations performed in the paper.

4.6 | Primary studies focusing on priority-based servicing

In this subsection, we investigated and analyzed the primary articles that focus on priority-based servicing in the field of AQTMCC. Of the studied articles, 4% fall in this category.

Banerjee et al[92] proposed a model in which VMs were modeled as service centers using $M/E_k/1$ and $M/E_k/2$ models. The authors studied a priority-based service time distribution method using the Erlang distribution for K-phases. Experiments showed that the $M/E_K/1$ model produced better results compared to the M/M/1 model and that $M/E_K/2$ has shown better throughput compared to M/M/2 in terms of average queue length and average waiting time. Dakshayini and Guruprasad[23] proposed a new scheduling algorithm based on a priority and admission control scheme. They considered an M/G/c queue model for a priority-based group of requests in a cloud environment; all the queues together make a queuing network. The policy with the proposed cloud architecture has achieved a very high (99%) service completion rate with guaranteed QoS over the traditional scheduling policy, which did not consider the priority and admission control techniques. Brandwajn and Begin[93] presented a simple approximate solution for preemptive-resume queues. They found that the proposed approximation provides a relatively simple and generally accurate approach to preemptive-resume queues with larger numbers of servers and general distributions of service and inter-arrival times. The Ph/Ph/c/N queuing system was used. In their approach, priority levels were solved one at a time in the order of decreasing priority. To deal with general service time distributions at each priority, the level used a reduced state. They found that the proposed approach can be readily applied to multiserver queues with preemptive-restart priority levels.

We thoroughly investigated and analyzed the primary studies in the field of AQTMCC that focused on priority-based servicing. Our observations are summarized in Table 11. This analysis Table contains the names of the publishers, publication year, author names, the queuing models used in the study, queue disciplines, advantages, disadvantages, experimental platforms, journals/conferences publishing the paper, and the parameters and computations performed in the paper.

4.7 | Primary studies focusing on reliability

In this subsection, we investigated and analyzed the primary articles that focus on reliability in the field of AQTMCC. Of the studied articles, 4% fall in this category.

Dai et al[94] developed a cloud service reliability model and a novel evaluation algorithm. They first elaborated various types of possible failures in a cloud service; based on that, they developed a holistic reliability model using Markov models, queuing theory, and graph theory. They proposed a new algorithm to evaluate cloud service reliability based on the developed model. Mahato and Singh[95] studied the reliability of the grid transaction processing system considering time-out failure, blocking failure, matchmaking failure, network failure, software program failure, resource failure, and deadline-miss failure using queuing theory; the M/M/c model is used for modeling. Li et al[96] formulated the service reliability

TABLE 10 An overview of existing primary studies focusing on resource management

Year	Publisher	Authors	Queuing Models (Discipline)	Advantages	Disadvantages	Experimental Platforms	Parameters/Computations
2018	Springer	Vakilinia and Cheriet [91]	Semi–birth-death	• Scheduling of heterogeneous workload is one challenging issue for cloud owners	• Not considering SLA contract	• Numerical analysis and simulation	• Service time • Blocking probability • Task failure rate
2016	IEEE	Shi et al [81]	M/M/C (FCFS)	• Increasing performance • Optimizing resource provisioning • Minimizing the number of servers to host service	• Not considering the heterogeneity of servers • Not considering the cost of service placement	• Using a custom simulator written in Python	• Arrival rate • Service rate • Number of servers required to host services • Maximum number of requests
2016	Springer	Song et al [90]	M/G/m/m + r (FCFS)	• Introducing deadline concept • Comparing analytical and simulation results	• Did not include quality of experience, media playback quality, and media service profiling	• Real environment (EC2)	• Resource utilization • Cost optimization • Waiting time
2014	Elsevier	Nan et al[88]	M/M/1, M/M//1, M/Hm/1 (FCFS)	• Minimizing response time and resource cost • Guaranteeing QoS • Considering heterogeneity	• No assurance for keeping QoS at a reasonable level	• Using practical parameters of Windows Azure	• Arrival rate • Service time • Response time • Resource cost
2013	IEEE	Xiong and He[84]	M/M/C (FCFS)	• Minimizing the mean end-to-end delay • Minimizing energy consumption • Considering multiple-server queue	• Not considering the heterogeneity of servers • Considering the same costs for all servers	• Numerical experiments • Using the Arena tool to simulated the proposed model	• Arrival rate • Service rate • End-to-end delay • Immediate service • Energy consumption • Cluster availability
2012	IEEE	Ellens et al[82]	M/M/C/C (FCFS)	• Periodization of requests in different classes • Decreasing queue length • Servicing according to SLA	• Assuming that different priority classes have the same average process time • Not considering batch arrivals • Not considering resource cost	• Numerical analysis and simulation	• Arrival rate • Service rate • Mean service time • Rejection probabilities • Number of reserved resources
2012	Elsevier	Casalicchio and Silvestri [85]	M/M/C, M/G/1 (FCFS)	• Dynamic resource provisioning • Automatic QoS-aware provisioning • Reducing resource cost	• Not considering the heterogeneity of servers	• Implementing a test bed by means of the Amazon EC2 infrastructure	• Arrival rate • Service rate • CPU utilization • Minimum number of VMs needed for service • Response time • VM allocation cost
2012	IEEE	Nan et al[89]	M/M/S (FCFS)	• Minimizing response time • Minimizing resource cost • Considering priority service scheme	• Not considering the delay requirement of applications • Not considering the heterogeneity of servers	• Employing the configuration and pricing rate of Windows Azure in simulation	• Arrival rate • Service rate • Response time • Resource cost
2011	IEEE	Nan et al [87]	M/M/1, M/G/1 (FCFS)	• Minimizing response time and cost • Considering single and multiple classes	• Not scalable • The model is not realistic • Not considering the priority service scheme	• Microsoft Azure in simulation	• Arrival rate • Service rate • Response time • Resource cost
2009	ACM	Hu et al[86]	M/M/C (FCFS)	• Requiring the number of servers • Increasing throughput • Reducing waiting time for short jobs	• Considering only two classes • No implementation in a real-cloud platform • No simulation	• Numerical analysis using some examples	• Service rate • Probability that the system is empty • Traffic intensity • Response time • The smallest number of needed servers
2008	IEEE	Xiong and Perros[83]	M/M/1, M/M/1/B (FCFS)	• Considering SLA in resource allocation • Minimizing the total cost of computer resources • Satisfying QoS defined is SLA	• Not considering QoS manager realistic workload of MapReduce clusters • Not considering priority-based requests	• Numerical analysis • Simulating the M/M/1/B queue using the Arena simulation tool	• Arrival rate • Service rate • Cluster utilization • Average cluster availability • Pack logs rate in the queue

TABLE 11 An overview of existing primary studies focusing on priority-based servicing

Year	Publisher	Authors	Queuing Models (Discipline)	Advantages	Disadvantages	ExperimentalPlatforms	Parameters/Computations
2017	Elsevier	Brandwajn and Begin[93]	Ph/Ph/c/N	• Avoiding starvation • Optimizing waiting time	• Used classical phase-type distributions, and no any phase-type	• Numerical analysis	• Arrival rate • Departure rate • Steady-state probability
2014	ACEEE	Banerjee et al[92]	M/Ek/1, M/Ek/2 (FCFS)	• Reducing queue length	• No implementation in a real-cloud environment		• Arrival rate • Service time • Queue length • Waiting time
2011	Foundation of Computer Science (FCS)	Dakshayini and Guruprasad[23]	M/G/C (FCFS)	• High service completion rate • Minimizing response time • Maximizing throughput	• Increasing overall service cost for the cloud • Not considering cross-cloud resources	• Numerical analysis and simulation	• Service time • Server utilization • Average time spent by a user in the system • Service cost • Total profit of servicing

TABLE 12 An overview of existing primary studies focusing on reliability

Year	Publisher	Authors	Queuing Models (Discipline)	Advantages	Disadvantages	Experimental Platforms	Parameters/Computations
2018	IEEE	Mahato and Singh[95]	M/M/c (FCFS)	• Considering time-out, blocking, and network and resource failure	• Analysis of a small case	• Numeric Analysis	• Transactions time • Reliability • Various types of failures
2017	Elsevier	Li et al[96]	M/M/Z/L (FCFS)	• Calculating service reliability	• Not considering parallel computations • Not considering node capacity	• Monte Carlo simulation	• Service reliability • Total no. of subtasks • Number of defects
2009	IEEE	Dai et al[94]	M/M/1 (FCFS)	• Considering many types of failures	• Service reliability not validated by simulation and real-life data • Lack of scalability	• Numerical analysis	• Overflow and time-out failure • Data source missing failure • Software, database, hardware, network, and link failures

model using the queuing theory and graph theory and provided its corresponding quantitative calculation and evaluation method based on the Monte Carlo method. In the proposed model, they considered various types of failures that influence service reliability.

We thoroughly investigated and analyzed the primary studies in the field of AQTMCC that focused on reliability. Our observations are summarized in Table 12. This analysis Table contains the names of the publishers, publication year, author names, the queuing models used in study, queue disciplines, advantages, disadvantages, experimental platforms, journals/conferences publishing the paper, and the parameters and computations performed in the paper.

5 | RESULTS

After synthesizing the data, we had the answers to our RQs, as presented in Sections 5.1 through 5.7. First of all, the heterogeneities of primary studies are shown. Some studies focused on homogeneous servers, but some of them focused on heterogeneous servers. Table 13 classified the primary studies in terms of heterogeneity or homogeneity.

5.1 | Answer to RQ1: *How many SLR, SMS, or regular reviews have there been since 2008 in the field of AQTMCC?*

With respect to RQ1, it may be a concern that we started our search at the beginning of 2008. Through a comprehensive search in the sources of Table 1, we recognized that there were not a significant number of papers in the AQTMCC area prior to 2008. Our study showed that during the period from January 2008 to 2018, there was not any SMS or SLR in the field of AQTMCC. However, there were a handful of regular surveys; five of them are shown in Table 5. Therefore, our paper is the first comprehensive secondary study that conducts a combination of SMS and SLR.

5.2 | Answer to RQ2: *What are the existing primary studies of AQTMCC, annual distribution, and their focus area?*

For answering RQ2, we found 71 articles by searching in the sources of Table 1. These articles were classified into seven categories based on their focus area, which are shown in Table 4. The Table shows that 26 articles out of the 71 studied articles focused on performance, and 10 articles out of them focused on QoS. By knowing that performance is a subset of QoS, we can say that 36 (26 + 10) articles out of 71 focused on QoS. Nine articles focused on workflow scheduling, nine articles focused on energy savings, 11 articles focused on resource management, three articles focused on priority-based servicing, and three articles focused on reliability. Figure 5 shows the distribution of primary articles per category. In the Figure, we showed the article frequencies of each category in the corresponding slice of the pie chart; the percentage of each category is also shown. It is observed in the Figure that a majority of the primary articles focused on performance (37%), 15% focused on resource management, 14% focused on QoS, 13% focused on workflow scheduling, 13% focused on energy savings, 4% focused on priority-based servicing, and 4% focused on reliability.

Figure 6 shows the annual distribution of the studied publications. We observe in the Figure the article frequencies of each year in the corresponding slice of the pie chart. The annual percentages of publications are as follows: 2% in 2008, 6% in 2009, 5% in 2010, 7% in 2011, 9% in 2012, 9% in 2013, 11% in 2014, 17% in 2015, 16% in 2016, 6% in 2017, and 12% in 2018. The Figure shows an increasing trend of published papers in the field of AQTMCC, which indicated the trend of cloud computing.

5.3 | Answer to RQ3: *What are the publication statistics and venue of the existing primary studies on AQTMCC in the literature?*

For answering RQ3, we refer to Figures 7 and 8. In both Figures, we showed article frequencies in the corresponding slice of the pie chart. In Figure 7, we see that a majority of the articles are published by the IEEE; the publication statistics are as follows: 39% of the articles are published by the IEEE, 23% are published by Springer, 14% are published by Elsevier, 6% are published by ACM, 4% are published by ProQuest, and 14% are published by other publishers. In Figure 8, we see the venue types; 40 articles out of 71 were published in journals, 30 articles out of 71 were presented at conferences, and one article was presented in a workshop. In other words, 56% of the articles were published in journals, 42% were presented at conferences, and 2% were presented in workshops.

TABLE 13 Classification of primary studies in terms of heterogeneity

Studies Dealing With Heterogeneous Servers	Studies Dealing With Homogeneous Servers
34,35,37,40,41,72,75,94, 46,47,51,60,68,71,76,90,91	5,22,31-33,36,37,39,54-58,63-66,73,81,82 17,23,42-45,48-50,59,67,74,77,83-89,92 20,52,53,61,62,69,70,78-80,93,95,96

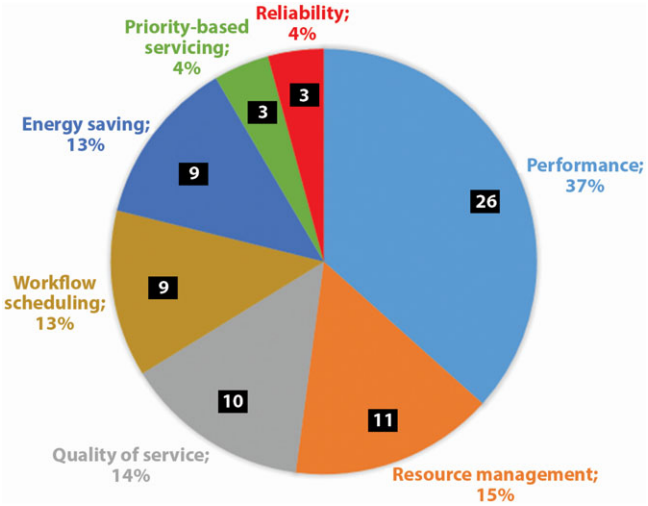

FIGURE 5 Distribution of articles per category

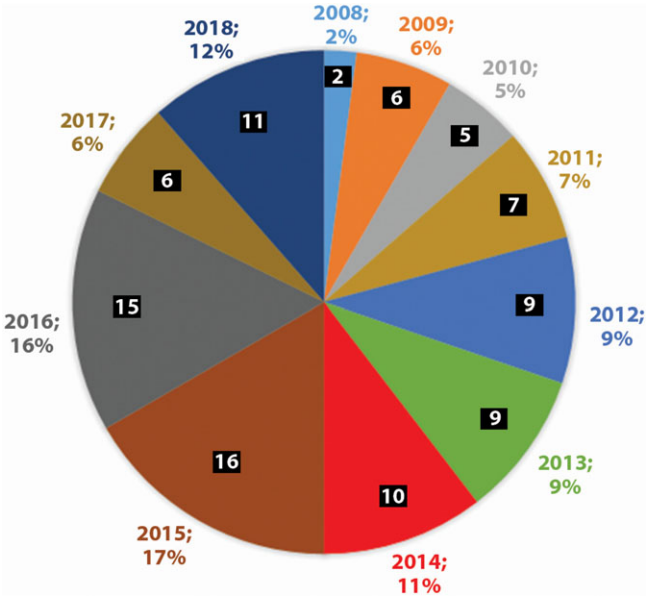

FIGURE 6 Annual distribution of publications from 2008 to 2018

5.4 | Answer to RQ4: *What kinds of queuing models are used in the field of AQTMCC and which one is frequently used?*

For answering RQ4, we refer to Figures 9 and 10. For more clarity, a diagram is presented in Figure 9, a taxonomy of AQTMCC, that reveals which queue models were applied in modeling cloud computing in seven categories, namely, QoS, performance, workflow scheduling, energy savings, resource management, priority-based service, and reliability. Figure 10 reveals the queuing models and their frequencies in the studied articles in the corresponding slice of the pie chart; the percentage of each model is also shown. In our study, we observed that some articles use more than one queuing model. For example, Murugan et al[54] used the models M/M/1 and M/M/m, whereas Nan et al[88] used the queuing models M/M/1, M/M//1, and M/H_m/1. In Figure 10, we see that the M/M/c model is used more than the other models (it was used in 26 papers). Of the articles, 39% used the M/M/c model, 27% used the M/M/1 model, 11% used the M/G/1 model, 5% used the M/M/∞ model, 5% used the M/G/m model, and 13% used other models.

5.5 | Answer to RQ5: *What experimental platforms are used by the researchers for analysis and evaluation of the primary studies?*

Figure 11 helps us answer RQ5. The required information to the Figure has been extracted from Tables 6-12. In the Figure, we classified the experimental platforms into six categories: numerical analyses; CloudSim tool; discrete-event simulators; written programs by the authors in languages such as C++, Java, and MATLAB; available software packages such as Arena; and real implementations. We observed that in

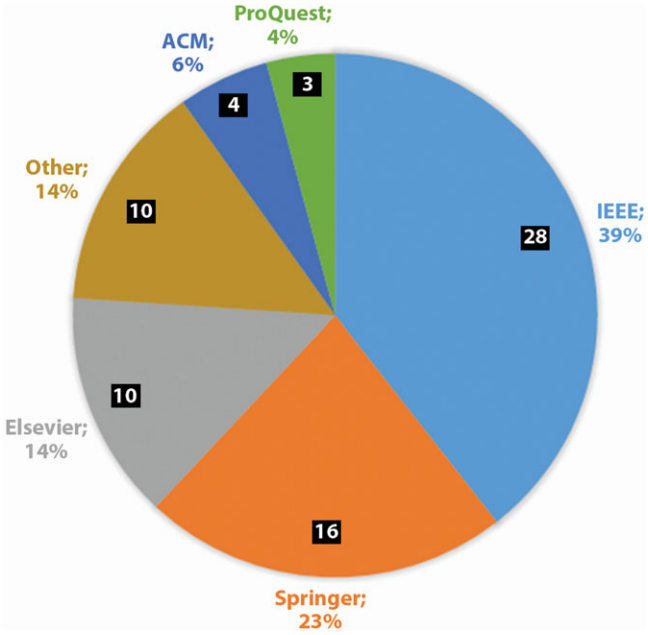

FIGURE 7 Studies by publisher

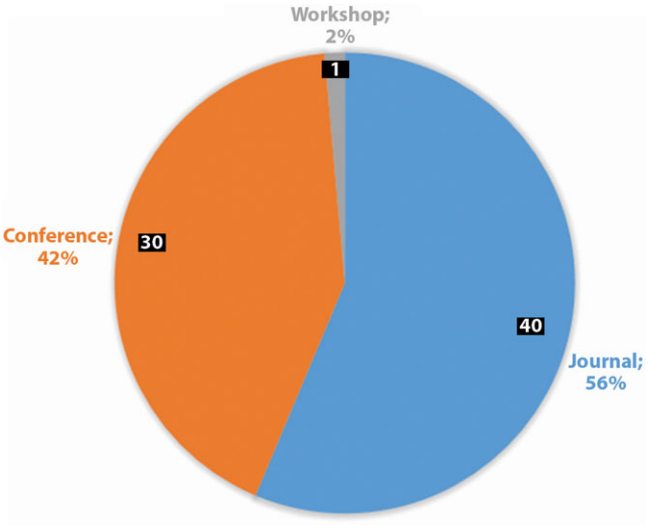

FIGURE 8 Studies by venue

some articles, more than one platform was used; for example, Khazaei et al[41] in Table 6 used Maple for solving balance equations and built a discrete-event simulator using Artifex for simulation. The article frequencies of each category are shown in the corresponding slice of the pie chart. We see that, of the articles, 41% used numerical analysis, 17% evaluated their model by real implementation, 13% used a discrete-event simulator, 13% used available software packages, 8% used the CloudSim tool, and 8% of the authors wrote programs for model evaluation.

5.6 | Answer to RQ6: *What queue disciplines were used in AQTMCC?*

The queue discipline is the method in which the customers are selected by the servers, or vice versa.[4] Tables 6-12 reveal that nearly all the queuing models in AQTMCC used the FCFC discipline, except that in the work of Mary and Saravanan,[32] which used a general discipline. Dakshayini and Guruprasad,[23] Murugan et al,[54] Li,[65] and Banerjee et al[92] placed the requests based on their priorities in separate queues, whereas in each queue, the FCFS discipline was used to process the requests.

5.7 | Answer to RQ7: *What are the open issues of AQTMCC research studies?*

Based on the characteristics of the primary AQTMCC studies, we want to find out the open issues that would deserve more investigation in the future and some potential directions to tackle these issues.

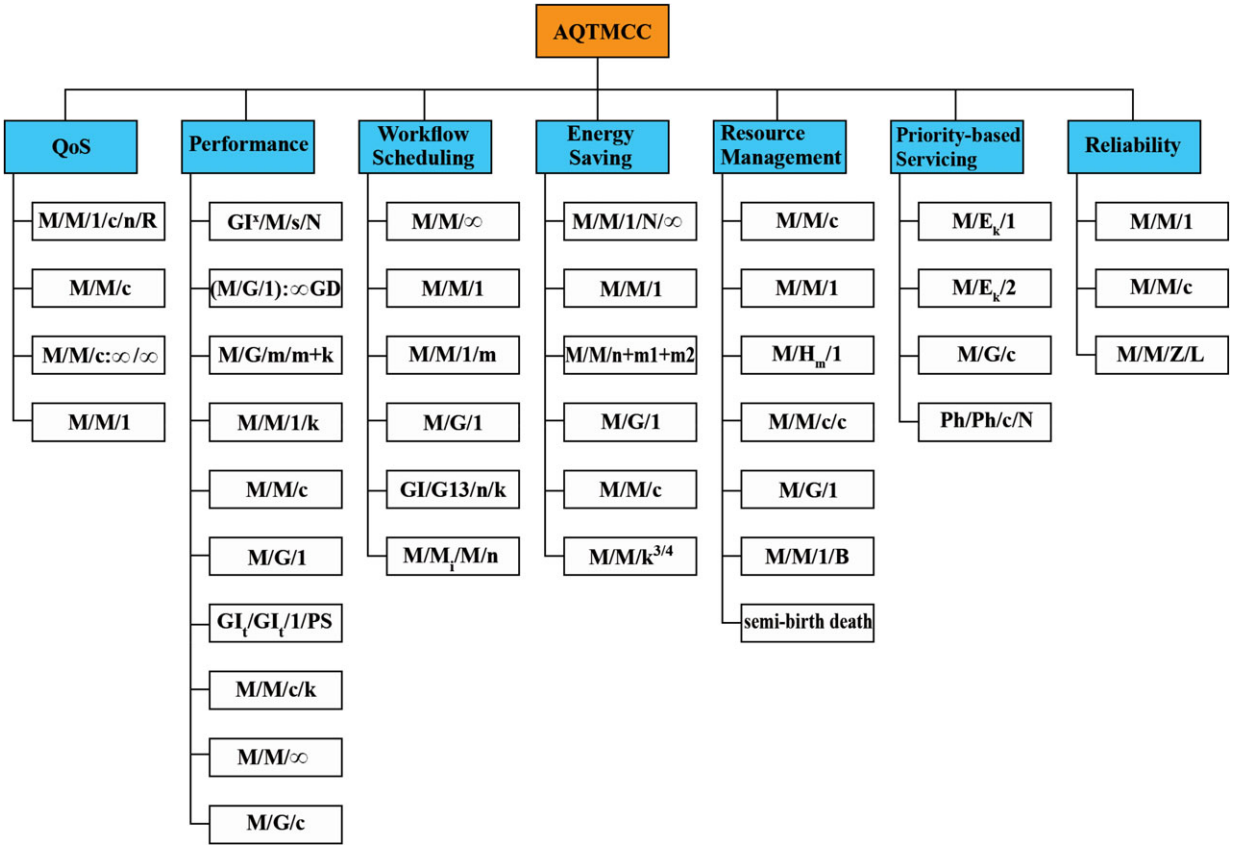

FIGURE 9 A taxonomy of queue models for AQTMCC

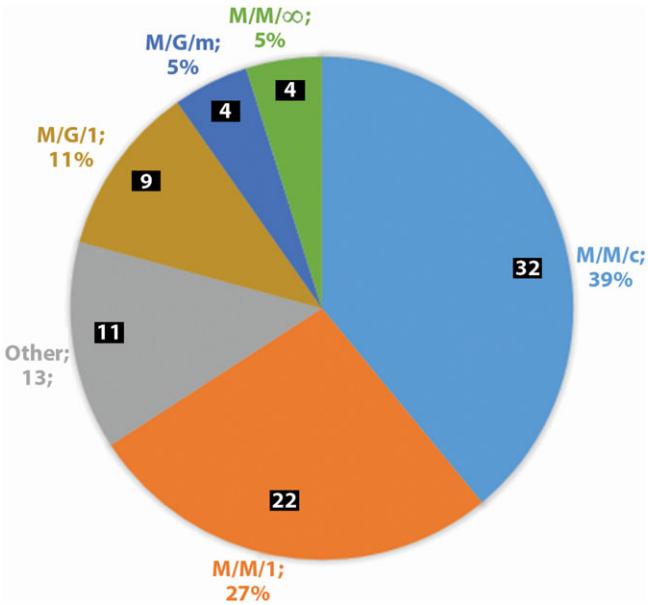

FIGURE 10 Queuing models used in studies

Our answer to RQ2 states that slightly more than half of the existing primary AQTMCC studies addressed performance and QoS. While these topics are important from the perspective of a cloud user and provider, topics such as reliability and cost of service for cloud users and providers were not significantly addressed. Security is another issue, which is missing in the studies of AQTMCC.

Recently, cloud data centers have been growing increasingly, and energy consumption and carbon emission are becoming a challenge; in AQTMCC, a limited number of papers addressed the topic. Automatic QoS-aware resource management and automatic SLA-aware resource

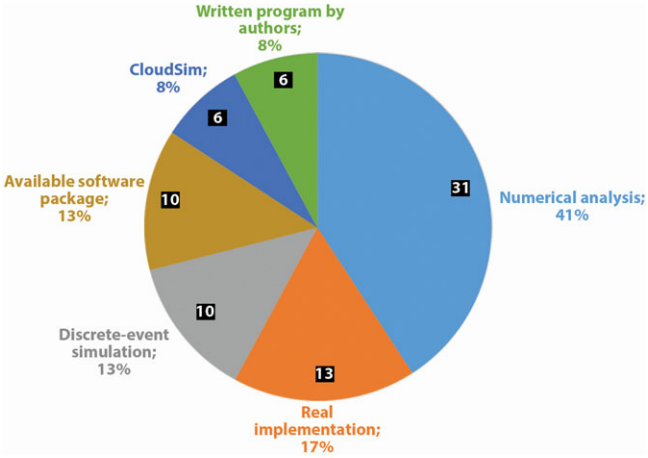

FIGURE 11 Experimental platforms used in studies

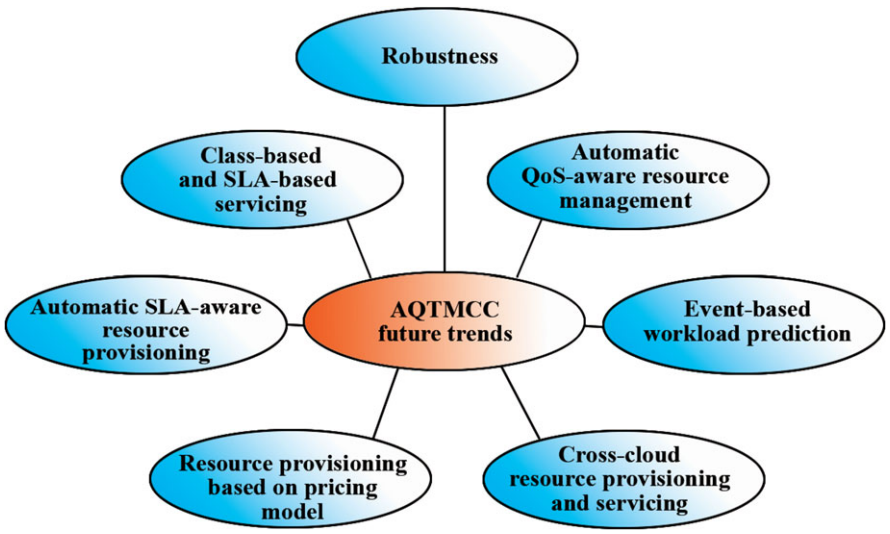

FIGURE 12 AQTMCC open issues and future works

management are other issues the solutions for which are still rare. Another issue with AQTMCC is that only a limited number of primary studies implemented their queuing models in a real-life cloud environment, whereas the simulation environment is far from reality. One of the emerging issues is cross-cloud resource provisioning, which was not addressed in the primary studies of AQTMCC.

It seems that the general trend in cloud computing is toward making use of services from multiple-cloud providers. Therefore, investigating cross-cloud resource provisioning and servicing for satisfying customers by applying a queuing system is a valuable research topic. Due to the uncertainty of the cloud environment, including the varying nature of customer requests, we invite the researchers for modeling cloud computing using a queuing system for robustness. By reviewing and analyzing the literature, it has been observed that there is not any independent technique that addresses all issues involved in green computing. Hence, another fascinating point of future study would be to investigate metrics such as energy savings, carbon emission, service cost, and settle time (wait time in queue plus service time) together. The literature review revealed that class-based and SLA-based servicing is less considered, whereas it is possible to do that using a queueing system. Resource provisioning and management based on a pricing model, automatic QoS-aware resource management, and SLA-aware resource management are other important topics requiring attention. Figure 12 shows the AQTMCC open issues and future works.

6 | CONCLUSIONS

In this paper, we have presented the results of an SMS combined with SLR of the existing studies of AQTMCC. The results shed light on the real history of AQTMCC. We classified the primary studies based on their investigated parameters. We found that a majority of the primary

studies of AQTMCC focused on performance (26 of 71 articles). Moreover, we identified the distributions over popular publishers in this field. We found that famous publishers published the greatest fraction of articles; 39% of the articles are published by the IEEE. The paper provides a taxonomy of queuing models in cloud computing, and it determines the future path of AQTMCC and the need for conducting more research on AQTMCC. We draw the future line of research as robustness, automatic QoS-aware and SLA-aware resource provisioning, event-based workload prediction, cross-cloud resource provisioning and servicing, provisioning based on a pricing model, and class-based and SLA-based servicing.

This study has some limitations. Firstly, it only surveyed articles published from 2008, which were extracted based on some search strings. Secondly, this study tried to investigate the articles published by famous publishers. There might be other publications, conferences, workshops, and symposiums that may provide more comprehensive articles related to AQTMCC. Lastly, non-English publications were excluded from the study.

ORCID


Amir Masoud Rahmani https://orcid.org/0000-0001-8641-6119

REFERENCES

1. Alam MI, Pandey M, Rautaray SS. A comprehensive survey on cloud computing. *Int J Inf Technol Comput Sci*. 2015;7(2):68-79.
2. Mell P, Grance T. The NIST definition of cloud computing. NIST Special Publication 800-145. 2011.
3. Zhang Q, Cheng L, Boutaba R. Cloud computing: state-of-the-art and research challenges. *J Internet Serv Appl*. 2010;1:7-18.
4. Harchol-Balter M. *Performance Modeling and Design of Computer Systems*. Cambridge, UK: Cambridge University Press; 2013.
5. Vilaplana J, Solsona F, Teixidó I, Mateo J, Abella F, Rius J. A queuing theory model for cloud computing. *J Supercomput*. 2014;69(1):492-507.
6. Petersen K, Vakkalanka S, Kuzniarz L. Guidelines for conducting systematic mapping studies in software engineering: an update. *Inf Softw Technol*. 2015;64:1-18.
7. Kitchenham BA, Budgen D, Brereton OP. Using mapping studies as the basis for further research—a participant-observer case study. *Inf Softw Technol*. 2011;53:638-651.
8. Qu C, Calheiros RN, Buyya R. Auto-scaling web applications in clouds: a taxonomy and survey. *ACM Comput Surv*. 2018;51(4). Article No. 73.
9. Chen T, Bahsoon R, Yao X. A survey and taxonomy of self-aware and self-adaptive cloud autoscaling systems. *ACM Comput Surv*. 2018;51(3). Article No. 61.
10. Aslanpour MS, Ghobaei-Arani M, Nadjaran Toosi A. Auto-scaling web applications in clouds: a cost-aware approach. *J Netw Comput Appl*. 2017;95:26-41.
11. Kitchenham B, Brereton OP, Budgen D, Turner M, Bailey J, Linkman S. Systematic literature reviews in software engineering – a systematic literature review. *Inf Softw Technol*. 2009;51:7-15.
12. Carvalho JFS, Neto PAMS, Garica VC, Assad RE, Durao F. *A Systematic Mapping Study on Cloud Computing*. Recife, Brasil: Universidade Federal de Pernambuco; 2013.
13. Kitchenham B, Brereton P, Budgen D. The educational value of mapping studies of software engineering literature. In: Proceedings of ACM/IEEE International Conference on Software Engineering; 2010; Cape Town, South Africa.
14. Petersen K, Feldt R, Mujtabal S, Mattsson M. Systematic mapping studies in software engineering. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering; 2008; Bari, Italy.
15. Jula A, Sundararajan E, Othman Z. Cloud computing service composition: a systematic literature review. *Expert Syst Appl*. 2014;41(8):3809-3824.
16. Rimal BP, Jukan A, Katsaros D, Goeleven Y. Architectural requirements for cloud computing systems: an enterprise cloud approach. *J Grid Comput*. 2011;9(1):3-26.
17. Anupama A, Keerthi GS. Using queuing theory the performance measures of cloud with infinite servers. *Int J Comput Sci Eng Technol*. 2014;5(1):17-21.
18. Daraghmi EY, Yuan S-M. A small world based overlay network for improving dynamic load-balancing. *J Syst Softw*. 2015;107:187-203.
19. Armbrust M, Fox A, Griffith R, et al. A view of cloud computing. *Commun ACM*. 2010;53(4):50-58.
20. Pal S, Pattnaik PK. A simulation-based approach to optimize the execution time and minimization of average waiting time using queuing model in cloud computing environment. *Int J Elect Comput Eng*. 2016;6(2):743-750.
21. Calheiros RN, Ranjan R, De Rose CAF, Buyya R. *CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services*. Technical Report. Melbourne, Australia: Grid Computing and Distributed Systems Laboratory, University of Melbourne; 2009.
22. Ait-Salaht F, Castel-Taleb H. Stochastic bounding models for performance analysis of clouds. Paper presented at: 15th IEEE International Conference on Computer and Information Technology (CIT-2015); 2015; Liverpool, UK.
23. Dakshayini M, Guruprasad HS. An optimal model for priority based service scheduling policy for cloud computing environment. *Int J Comput Appl*. 2011;32(9):23-29.
24. Mo J. *Performance Modeling of Communication Networks with Markov Chains*. San Rafael, CA: Morgan & Claypool; 2012.
25. Murugesan R, Elango C, Kannan S. A status report on resource allocation in cloud computing using queuing theory. *Int J Adv Res Comput Eng Technol*. 2014;3(11):3603-3608.
26. Zhang J, Huang H, Wang X. Resource provision algorithms in cloud computing: a survey. *J Netw Comput Appl*. 2016;64:23-42.
27. Manvi SS, Shyam GK. Resource management for Infrastructure as a Service (IaaS) in cloud computing: a survey. *J Netw Comput Appl*. 2014;41:424-440.

28. Lorido-Botran T, Miguel-Alonso J, Lozano JA. A review of auto-scaling techniques for elastic applications in cloud environments. *J Grid Comput Manuscr*. 2014;12(4):559-592.
29. Santhi K, Saravanan R. A survey on queueing models for cloud computing. *Int J Pharmacy Technol*. 2016.
30. Kitchenham BA, Budgen D, Brereton OP. The value of mapping studies—a participant-observer case study. In: Proceedings of the 14th International Conference on Evaluation and Assessment in Software Engineering (EASE'10); 2010; Keele, UK.
31. Ever E. Performability analysis of cloud computing centers with large numbers of servers. *J Supercomput*. 2017;73:2130-2156.
32. Mary NA, Saravanan K. Performance factors of cloud computing data centers using [(M/G/1): (∞/GDMODEL)] queuing systems. *Int J Grid Comput Appl*. 2013;4(1):1-9.
33. Chang X, Wang B, Muppala JK, Liu J. Modeling active virtual machines on IaaS clouds using an *M/G/m/m+K* queue. *IEEE Trans Serv Comput*. 2016;9(3):408-420.
34. Bai W-H, Xi J-Q, Zhu J-X, Huang S-W. Performance analysis of heterogeneous data centers in cloud computing using a complex queuing model. *Math Probl Eng*. 2015.
35. Liu X, Li S, Tong W. A queuing model considering resources sharing for cloud service performance. *J Supercomput*. 2015;71(11):4042-4055.
36. Varma PS, Satyanarayana A, Sundari MR. Performance analysis of cloud computing using queuing models. Paper presented at: International Conference on Cloud Computing Technologies, Applications, and Management (ICCCTAM); 2012; Dubai, United Arab Emirates.
37. Khazaei H, Misic J, Misic VB. Modelling of cloud computing centers using M/G/m queues. Paper presented at: 31st International Conference on Distributed Computing Systems Workshops; 2011; Minneapolis, MN.
38. Goswami V, Patra SS, Mund GB. Performance analysis of cloud with queue-dependent virtual machines. Paper presented at: 1st International Conference on Recent Advances in Information Technology (RAIT); 2012; Dhanbad, India.
39. Raei H, Yazdani N, Shojaee R. Modeling and performance analysis of cloudlet in mobile cloud computing. *Perform Eval*. 2017;107:34-53.
40. Vakilinia S, Ali MM, Qiu D. Modeling of the resource allocation in cloud computing centers. *Comput Netw*. 2015;91:453-470.
41. Khazaei H, Misic J, Misic VB. Performance analysis of cloud computing centers using M/G/m/m+r queuing systems. *IEEE Trans Parallel Distrib Syst*. 2011;23(5):936-943.
42. Khojasteh H, Misic J, Misic V. Prioritization of overflow tasks to improve performance of mobile cloud. *IEEE Trans Cloud Comput*. 2016.
43. Nguyen BM, Tran D, Nguyen G. Enhancing service capability with multiple finite capacity server queues in cloud data centers. *Clust Comput*. 2016;19(4):1747-1767.
44. Keller M, Karl H. Response time-optimized distributed cloud resource allocation. In: Proceedings of the 2014 ACM SIGCOMM Workshop on Distributed Cloud Computing (DCC); 2014; Chicago, IL.
45. RahimiZadeh K, AnaLoui M, Kabiri P, Javadi B. Performance modeling and analysis of virtualized multi-tier applications under dynamic workloads. *J Netw Comput Appl*. 2015;56:166-187.
46. Sun P, Wu D, Qiu X, Luo L, Li H. Performance analysis of cloud service considering reliability. Paper presented at: IEEE International Conference on Software Quality, Reliability and Security Companion; 2016; Vienna, Austria.
47. Liu X, Tong W, Zhi X, Fu ZhiRen F, WenZhao L. Performance analysis of cloud computing services considering resources sharing among virtual machines. *J Supercomput*. 2014;69(1):357-374.
48. Shi Y, Huang J, Zhao X, Liu L, Liu S, Cui L. Integrating theoretical modeling and experimental measurement for soft resource allocation in multi-tier web systems. Paper presented at: IEEE International Conference on Web Services; 2016; San Francisco, CA.
49. Khazaei H, Misic J, Misic VB. A fine-grained performance model of cloud computing centers. *IEEE Trans Parallel Distrib Syst*. 2013;24(11):2138-2147.
50. Akingbesote AO, Adigun MO, Sanjay M, Ajayi IR. Performance analysis of non-preemptive priority with application to cloud E-marketplaces. Paper presented at: IEEE 6th International Conference on Adaptive Science & Technology (ICAST); 2014; Ota, Nigeria.
51. Fakhrolmobasheri S, Ataie E, Movaghar A. Modeling and evaluation of power-aware software rejuvenation in cloud systems. *Algorithms*. 2018;11(10):160.
52. Hanini M, El Kafhali S. Cloud computing performance evaluation under dynamic resource utilization and traffic control. In: Proceedings of the 2nd International Conference on Big Data, Cloud and Applications; 2017; Tetouan, Morocco.
53. El Kafhali S, Salah K. Modeling and analysis of performance and energy consumption in cloud data centers. *Arabian J Sci Eng*. 2018;43:7789-7802.
54. Murugan M, Aminu H, Subramanian G. Mathematical analysis on quality of service in cloud servers. *Int J Enhanc Res Sci Technol Eng*. 2015;4(10).
55. Kirsal Y, Ever YK, Mostarda L, Gemikonakli O. Analytical modelling and performability analysis for cloud computing using queuing system. Paper presented at: IEEE/ACM 8th International Conference on Utility and Cloud Computing; 2015; Limassol, Cyprus.
56. Xiong K, Perros H. Service performance and analysis in cloud computing. *IEEE World Conf Serv*. 2009;693-700.
57. Khomonenko AD, Gindin SI, Modher KM. A cloud computing model using multi-channel queuing system with cooling. Paper presented at: XIX IEEE International Conference on Soft Computing and Measurements (SCM); 2016; St. Petersburg, Russia.
58. Rajendran VV, Swamynathan S. Queuing model for improving QoS in cloud service discovery. In: *Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA) 2015*. New Delhi, India: Springer; 2016:647-656.
59. Vilaplana J, Solsona F, Abella F, Filgueira R, Rius J. The cloud paradigm applied to e-Health. *BMC Med Inform Decis Mak*. 2013;35(13):1-10.
60. Vilaplana J, Solsona F, Teixido I. A performance model for scalable cloud computing. In: Proceedings of the 13th Australasian Symposium on Parallel and Distributed Computing; 2015; Sydney, Australia.
61. Cho Y, Ko YM. Stabilizing the virtual response time in single-server processor sharing queues with slowly time-varying arrival rates. 2018. arXiv preprint arXiv:1811.01611.
62. Melikov AZ, Rustamov AM, Sztrik J. Queuing management with feedback in cloud computing centers with large numbers of web servers. In: *Distributed Computer and Communication Networks 21st International Conference, DCCN 2018, Moscow, Russia, September 17-21, 2018, Proceedings*. Cham, Switzerland: Springer; 2018:106-119.

63. He T-Q, Cai L-J, Deng Z-Y, Wang X, Tao M. Queuing-oriented job optimization scheduling in cloud mapreduce. In: *Advances on P2P, Parallel, Grid, Cloud and Internet Computing: Proceedings of the 11th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2016) November 5-7, 2016, Soonchunhyang University, Asan, Korea*. Vol 1. Cham, Switzerland: Springer; 2016:435-446.
64. Eisa M, Esedemy EI, Rashad MZ. Enhancing cloud computing scheduling based on queuing models. *Int J Comput Appl*. 2014;85(2).
65. Li L. An optimistic differentiated service job scheduling system for cloud computing service users and providers. Paper presented at: Third International Conference on Multimedia and Ubiquitous Engineering; 2009; Qingdao, China.
66. Dutta K, Guin RB, Banerjee S, Chakrabarti S, Biswas U. A smart job scheduling system for cloud computing service providers and users: Modeling and simulation. Paper presented at: 1st International Conference on Recent Advances in Information Technology (RAIT); 2012; Dhanbad, India.
67. Rashidi S, Sharifian S. A hybrid heuristic queue based algorithm for task assignment in mobile cloud. *Future Gener Comput Syst*. 2017;68:331-345.
68. Peng Z, Cui D, Zuo J, Li Q, Xu B, Lin W. Random task scheduling scheme based on reinforcement learning in cloud computing. *J Cluster Comput*. 2015;18(4):1595-1607.
69. Srivastava R. Analysis of job scheduling algorithm for an E-business model in a cloud computing environment via GI/G/3/N/K queuing model. *Int J Adv Technol*. 2012;215-229.
70. Sundararaj V. Optimal task assignment in mobile cloud computing by queue based ant-bee algorithm. *Wirel Pers Commun*. 2018;1:173-197.
71. Narman HS, Hossain MS, Atiquzzaman M, Shen H. Scheduling internet of things applications in cloud computing. *Annals Telecommun*. 2017;72(1-2):79-93.
72. Liao D, Li K, Gang Sun G, Anand V, Gong Y, Tan Z. Energy and performance management in large data centers: A queuing theory perspective. Paper presented at: 2015 International Conference on Computing, Networking and Communications (ICNC); 2015; Garden Grove, CA.
73. Bi J, Yuan H, Tan W, Li BH. TRS: temporal request scheduling with bounded delay assurance in a green cloud data center. *Inf Sci*. 2016;360:57-72.
74. Akbari E, Cug F, Patel H, Razaque A. Incorporation of weighted linear prediction technique and M/M/1 queuing theory for improving energy efficiency of cloud computing datacenters. Paper presented at: 2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT); 2016; Farmingdale, NY.
75. Cordeschi N, Shojafar M, Amendola D, Baccarelli E. Energy-efficient adaptive networked datacenters for the QoS support of real-time applications. *J Supercomput*. 2015;71(2):448-478.
76. Cheng C, Li J, Wang Y. An energy-saving task scheduling strategy based on vacation queuing theory in cloud computing. *Tsinghua Sci Technol*. 2015;20(1):28-39.
77. Shi Y, Jiang X, Ye K. An energy-efficient scheme for cloud resource provisioning based on CloudSim. Paper presented at: IEEE International Conference on Cluster Computing; 2011; Austin, TX.
78. Ghamkhari M, Mohsenian-Rad H. Energy and performance management of green data centers: a profit maximization approach. *IEEE Trans Smart Grid*. 2013;4(2):1017-1025.
79. Balde F, Elbiaze H, Gueye B. GreenPOD: Leveraging queuing networks for reducing energy consumption in data centers. Paper presented at: 2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN); 2018; Paris, France.
80. Chunxia Y, Shunfu J. An energy-saving strategy based on multi-server vacation queuing theory in cloud data center. *J Supercomput*. 2018;74(12):6766-6784.
81. Shi J, Dong F, Zhang J, Jin J, Luo J. Resource provisioning optimization for service hosting on cloud platform. IEEE 20th International Conference on Computer Supported Cooperative Work in Design; 2016; Nanchang, China.
82. Ellens W, Ivkovic M, Akkerboom J, Litjens R, van den Berg H. Performance of cloud computing centers with multiple priority classes. Paper presented at: IEEE Fifth International Conference on Cloud Computing; 2012; Honolulu, HI.
83. Xiong K, Perros H. SLA-based resource allocation in cluster computing systems. Paper presented at: 2008 IEEE International Symposium on Parallel and Distributed Processing; 2008; Miami, FL.
84. Xiong K, He Y. Power-efficient resource allocation in MapReduce clusters. Paper presented at: IFIP/IEEE International Symposium on Integrated Network Management (IM 2013); 2013; Ghent, Belgium.
85. Casalicchio E, Silvestri L. Mechanisms for SLA provisioning in cloud-based service providers. *Comput Netw*. 2013;57:795-810.
86. Hu Y, Wong J, Iszlai G, Litoiu M. Resource provisioning for cloud computing. In: Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research; 2009; Toronto, Canada.
87. Nan X, He Y, Guan L. Optimal resource allocation for multimedia cloud based on queuing model. Paper presented at: IEEE 13th International Workshop on Multimedia Signal Processing (MMSP); 2011; Hangzhou, China.
88. Nan X, He Y, Guan L. Queuing model based resource optimization for multimedia cloud. *J Vis Commun Image Represent*. 2014;25:928-942.
89. Nan X, He Y, Guan L. Optimal resource allocation for multimedia cloud in priority service scheme. Paper presented at: 2012 IEEE International Symposium on Circuits and Systems (ISCAS); 2012; Seoul, South Korea.
90. Song B, Hassan MM, Alamri A, et al. A two-stage approach for task and resource management in multimedia cloud environment. *Comput Secur*. 2016;98(1-2):119-145.
91. Vakilinia S, Cheriet M. Preemptive cloud resource allocation modeling of processing jobs. *J Supercomput*. 2018;74(5):2116-2150.
92. Banerjee C, Kundu A, Agarwal A, Singh P, Bhattacharya SW, Dattagupta R. Priority based K-Erlang distribution method in cloud computing. *Int J Recent Trends Eng Technol*. 2014;10(1):1-11.
93. Brandwajn A, Begin T. Multi-server preemptive priority queue with general arrivals and service times. *Perform Eval*. 2017;115:150-164.
94. Dai Y-S, Yang B, Dongarra J, Zhang G. Cloud service reliability: modeling and analysis. Paper presented at: 15th IEEE Pacific Rim International Symposium on Dependable Computing; 2009; Shanghai, China.

95. Mahato DP, Singh RS. Reliability modeling and analysis for deadline-constrained grid service. Paper presented at: 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA); 2018; Krakow, Poland.

96. Li X, Liu Y, Kang R, Xiao L. Service reliability modeling and evaluation of active-active cloud data center based on the IT infrastructure. *Microelectron Reliab*. 2017;75:271-282.

How to cite this article: Jafarnejad Ghomi E, Rahmani AM, Qader NN. Applying queue theory for modeling of cloud computing: A systematic review. *Concurrency Computat Pract Exper*. 2019;31:e5186. https://doi.org/10.1002/cpe.5186