云环境下随机请求性能分析综述

王 爽 李小平 陈 龙
(东南大学计算机科学与工程学院 南京 211189)
(计算机网络和信息集成教育部重点实验室 南京 211189)

摘 要 如何在云服务中心为随机到达系统的用户请求选择并分配合适资源以最优化某些性能指标是云计算的关键问题之一.不同云计算场景下请求到资源的映射产生不同排队模型.现有研究中,用户请求具有泊松到达、一般随机到达等不同模式,具有不同忍耐程度、截止期等约束;服务资源通常为单队列或多队列的异构处理器,其服务方式服从指数或其它随机分布.这些模式和约束组合成多类复杂排队模型,如何为每类供需服务确定合适的处理器数量并合理调度以优化响应时间和功耗、租赁成本、系统能耗、服务提供商收益等不同目标是云计算的关键问题.针对具有忍耐程度、截止期等不同约束的泊松到达用户请求,本文分别研究单队列和多队列异构处理器的排队性能分析和调度优化方法.请求的调度规则有先到先服务和最早开始截止时间优先.根据云计算特性,本文提出性能分析的核心问题;根据实际应用场景,本文分析问题挑战,总结多种排队模型;结合相关系统排队模型,本文提出通用的云计算环境性能分析框架和相应的性能分析步骤;本文综述目前云环境下随机请求性能分析的研究进展,考虑成本、收益、响应时间和能耗等优化目标,比较多种排队模型优缺点;展望未来研究方向.

关键词 随机请求;排队论;排队模型;性能分析;云计算
中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2022.01241

Performance Analysis for Stochastic Requests in Cloud Computing: A Survey

WANG Shuang LI Xiao-Ping CHEN Long
(*School of Computer Science and Engineering, Southeast University, Nanjing* 211189)
(*Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing* 211189)

Abstract The selection and allocation of heterogeneous resources from service providers for stochastic requests from service consumers is one of the key problems in cloud computing. It is vital to select and allocate these various resources reasonably for these key problems so that the cloud service systems can have the best performance in cloud computing. Many stochastic requests arrive at the cloud system dynamically with different distributions in terms of different scenarios in cloud computing. The different arrival patterns of stochastic requests for different kinds of consumers follow different distributions such as Poisson distribution, and general distribution with different constraints which can be defined by different patience from consumers and deadlines of requests. The various service resources are usually heterogeneous servers that follow the exponential distribution for single queue in real scenarios or multiple queues for other real scenarios. Different complex queueing models are constructed in terms of different arrival patterns, service patterns and constraint combinations which are changed by different conditions. How to determine the number of servers, and scheduling requests reasonably to optimize response time and power consumption for cloud service systems, minimize rental cost for service providers, minimize system energy consumption for cloud providers, and maximize profit for

收稿日期:2020-04-29;在线发布日期:2021-07-21.本课题得到国家重点研发计划项目(2017YFB1400800)、国家自然科学基金(61872077)资助.**王 爽**,博士,讲师,主要研究方向为任务调度、性能分析、云计算、强化学习. E-mail: shuangwang@seu.edu.cn. **李小平**(通信作者),博士,教授,博士生导师,中国计算机学会(CCF)杰出会员,主要研究领域为云计算、服务计算. E-mail: xpli@seu.edu.cn. **陈 龙**,博士,讲师,主要研究方向为云计算、服务计算等.

service providers are a series of critical problems for performance analysis with stochastic requests in cloud computing. Different queueing models are produced since requests are mapped to resources in different scenarios which implies there are various queueing models for different real scenarios in cloud computing. The rules of dispatching requests are First Come First Served and the start due time first. Different kinds of systems are analyzed in terms of these two kinds of rules in cloud computing. According to the stochastic property in cloud computing, the key problems for performance analysis with stochastic requests are proposed and studied. Based on the real application scenarios, different kinds of queueing models are constructed and analyzed where the queue theory is used to predict the performance with stochastic requests for a series of dynamic systems in cloud computing. According to the key problems, a series of challenges are proposed and solved for performance analysis with stochastic requests on a series of dynamic cloud systems in cloud computing. The general queueing model framework is proposed which can represent and analyze various kinds of existing queueing models. And the performance analysis procedures are proposed in terms of the existing queueing models which can be used to analyze new stochastic dynamic systems in cloud computing. The objectives including response time, cost, profit and energy consumption are analyzed and compared, respectively. According to these optimization objectives, a lot of existing work related to performance analysis for stochastic requests in cloud computing is analyzed and compared. The performance analysis is investigated by comparing different queueing models in cloud computing which contain the arrival patterns, the service patterns and the constraints. According to the analysis and comparison of existing queueing models, a series of promising topics are discussed which is worth studying and can be studied in the future.

Keywords stochastic service requests; queue theory; queueing models; performance analysis; cloud computing

1 引 言

云计算环境中云服务提供商以服务形式为用户提供资源，云服务系统性能是服务提供商和用户共同关心的关键问题. 延迟和吞吐量是评价云计算服务系统质量(Quality of Service)的重要指标[1]，延迟反映系统的请求响应时间而吞吐量表明系统可处理的最大请求数. 服务提供商根据用户需求从资源拥有者租赁处理器，但用户需求的随机动态性使得租赁成本、收益和能耗等随之动态变化，如图 1 所示. 用户根据延迟预测决定请求是否加入当前系统，即用户请求量随机动态变化. 服务提供商单位时间内尽可能多地接收请求，接收的请求越多，系统吞吐量越高，吞吐量越高表明收益越大，但需要处理器运行速率较快；同时处理器越快，成本越大，能耗也越多. 分析云服务系统的性能以支持用户所关注的性能指标(如响应时间)可提升用户满意度，还可预测服务提供商成本、收益和能耗等性能指标. 因此，云服务系统性能分析是有效管理云服务中心的关键问题.

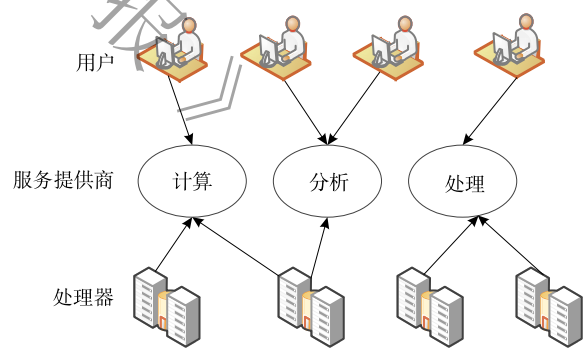

图 1 场景实例

性能分析通常是 NP 难的复杂问题[2]，云计算环境下请求(任务、作业)的到达方式、队列个数、排队方法、处理器类型、约束条件和目标函数等都是影响性能分析结果的重要因素，如随机请求到达的分布、到达时间间隔、同构/异构处理器、单队列/多队列等直接影响性能分析结果. 如何为动态异构资源选择合适处理器、确定处理器调度顺序、确定租赁模式并将请求分派到合适处理器进而提高系统性能是

一个很难的性能分析问题.随着实际应用中云服务规模和复杂度增加,与之对应的性能分析问题越来越困难.此外,性能分析通常与调度优化过程紧密相关,由于随机请求到处理器的不同映射生成不同系统性能指标,即可根据性能分析结果,设计优良的调度算法改善性能指标.

以阿里云呼叫中心为典型应用实例①,该系统为基于云端的呼叫中心服务,借助该服务企业以更低的成本获得更可靠和灵活的热线服务,提升企业的客户服务质量.呼叫中心平台上包括坐席、互动式语音应答、队列、录音等各项服务.呼叫中心的用户多样,用户提出的请求随机到达阿里云呼叫中心,如坐席(文中处理器)有空闲则请求得到处理,否则请求进入队列进行等待.由于请求到达随机特性和处理请求动态特性,考虑实际场景中请求、资源、目标等多种不同约束,如何评估呼叫中心性能是云环境下性能分析的关键问题.

针对这类问题,存在以下挑战:

(1)如何分析实际场景的不同到达模式、处理模式并基于排队规则构建合理排队模型;

(2)如何为随机到达系统的请求选择合适的异构处理器;

(3)如何将随机动态的请求分配到合适处理器;

(4)面向所考虑优化目标,如何基于已构建排队模型分析系统性能.

动态变化的云环境下的随机请求性能分析已引起广泛关注[3-9],不同学者从不同角度分析性能分析研究现状.基于请求到达的动态随机性,使用排队论[10]方法预测系统性能,该理论根据给定参数,构建对应排队模型,分析系统性能.Benedict 研究高性能计算(HPC)应用程序、资源、响应时间、能耗等性能分析问题[11].Ward 研究单队列模型在离散零件制造领域的应用[12].文献[13]综述自动化制造系统排队网络分析模型,在有特殊阻塞机制和有限排队位置约束下,Balsamo 等人得到阻塞排队网络解形式[13].李健强等人研究工作流模型性能分析方法,基于工作流网的定义,提出多维工作流网的概念[14].基于排队论对系统进行性能分析的研究中,文献[15]比较多队列和有限排队位置情况下系统性能指标.Schwarz 等人介绍分组性能分析方法分类模式[16].但在云计算场景中,请求密集且随机,资源按地理分布.不同用户约束不同,如截止期、成本和能耗.不同用户目标不同,如租赁成本[17]、功耗和请求响应时间[2].这些文献[11-14]主要研究制造领域、网络中不同类型请求的性能分析问题,而本文综述的云环境下随机请求性能分析并非针对特定场景的性能分析.由于云服务性能分析包括请求到达模式、处理器执行模式、队列类型及不同约束条件等多个环节,每个环节有多种可能情形,由此组合成大量复杂的性能分析问题.

为便于分析云计算系统的性能,本文采用与文献[10]相同的符号表示方法 A/B/C/C+X/Y+Z,其中 A 表示到达模式、B 表示执行模式、C 是队列系统中处理器数量、X 是最大排队容量,C+X 表示系统容量、Y 是排队规则、Z 表示用户忍耐程度.基于请求到达模式和处理器执行模式,本文构建单队列或多队列排队模型,分析系统性能,为系统确定合适处理器数量和类型,优化成本、收益、响应时间和能耗等.基于提出的排队框架,根据不同约束条件构建不同排队模型,本文有如下贡献:

(1)基于到达模式、执行模式、队列、目标对性能分析中已构建排队模型进行分类.

(2)基于各种各样的排队模型,构建统一排队模型框架.

(3)为分析各种不同排队模型,提出统一性能分析步骤.

(4)根据不同问题,分析和比较不同模型,提出新的研究问题.

2　问题分类

依据上述符号表示,影响云系统的关键因素包含请求到达模式、处理器执行模式、队列和目标.如表 1 所示用不同符号表示不同到达模式、执行模式、排队准则及忍耐程度.

表 1　排队符号 A/B/C/C+X/Y+Z

符号	特点	注释
A,B	M	指数分布
	G	一般分布
	Ph	位相分布
	M[d]	动态指数分布
C	1,2,…,∞	处理器数量
	N	N 个处理器
X	1,2,…,∞	排队容量
	R	R 个排队位置
Y	FCFS	先到先服务
	ESF	最早开始截止时间优先
Z	D	最大等待时间(MWT)
	θ	MWT 服从指数分布

① https://help.aliyun.com/document_detail/59970.html?spm=a2c4g.11186623.4.2.778377150Qj06X

2.1 模型分类

本文基于请求到达模式、处理器执行模式、队列和目标综述云环境下随机请求性能分析问题.

(1)到达模式由请求到达时间间隔决定,请求到达速率为一般分布和泊松分布,当到达速率是泊松分布时,请求到达时间间隔服从指数分布,否则请求到达时间间隔服从一般分布.

(2)服务时间表明处理器执行模式,服务时间分为一般分布和指数分布,当服务时间服从指数分布时,请求剩余时间服从相同指数分布,当服务时间服从一般分布时,请求剩余时间服从一般分布.

(3)在对随机请求性能分析过程中,现有队列分为:单队列[18]和多队列[19].单队列模型表明在整个系统中请求只有一个队列排队,而多队列模型表示请求在多个队列中进行排队.单队列模型通常用在小型的云服务中心,多队列模型通常用在网络中,平衡工作负载[19].

(4)成本、收益、能耗和请求响应时间是云计算中的重要目标.服务提供商关注成本和收益,他们希望成本最小化[20-21],收益最大化[22].云提供商关心能耗,而用户关注请求的响应时间.处理器在处理请求的过程中消耗能耗[23-24],能耗越多表明成本越大.根据动态电压调度[25],处理器单位时间内功率越大,服务处理速率越快,请求处理时间减少,响应时间减少.

排队模型在不同场景下由到达模式、处理模式和队列决定,云计算性能分析表现出多样性,由此对性能分析问题进行分类,如图2所示.根据图2可知,对云环境下随机请求性能分析时,请求单个或成批到达云系统,请求到达速率服从泊松分布或一般分布;由于处理器配置不同,云服务系统执行模式多样,处理时间分布服从指数分布或一般分布;基于不同云计算场景,构建单队列云服务系统或多队列云服务系统,优化性能目标,如成本、收益、能耗和响应时间.

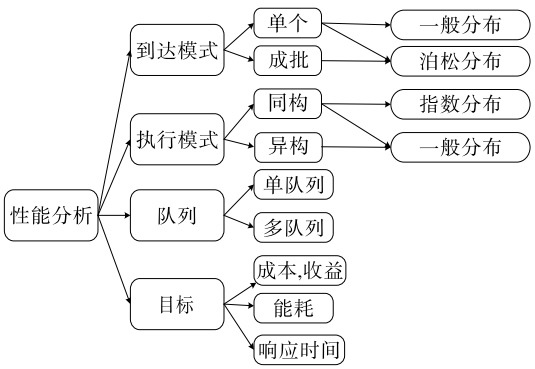

图2 模型分类

不同场景下,到达模式和执行模式服从不同分布,处理器数量和排队类型变化多样,常见的排队准则分为先到先服务、最早开始截止时间优先等.系统请求到达模式由请求个数和请求到达时间间隔的分布确定,请求到达分为单个到达和批量到达.请求到达分布包含泊松分布[3,22,26-33]和一般分布[4].如果请求到达服从泊松分布,这表明请求的到达时间间隔服从指数分布.尽管服务间隔服从确定约束的指数分布,状态转移过程具有规律性,但一般分布更适合一般场景,具有更好的灵活性,其复杂的状态转移过程使得性能分析过程更加复杂.在性能分析过程中,处理器可分为同构处理器[18,34]和异构处理器[31].同构处理器适用于一些特殊场景,而异构处理器适用于一般场景.处理器执行模式取决其服务时间分布,分为指数分布[18]和一般分布[35].根据不同队列个数,现有排队模型分单队列模型和多队列模型.不同云服务商关注目标不同,不同目标导致服务系统选择不同处理器,针对不同目标,性能分析过程不同.因此,本文比较多种目标比如成本[18,20-33,36-45]、收益[7,22,30,46]、能耗[5,47]和任务响应时间[5,19].服务提供商希望最小化成本、最大化收益、最小化能耗、最小化任务的响应时间或者同时优化这些目标.针对不同云计算场景,基于到达模式、执行模式、处理器数量、排队容量、排队准则、用户忍耐程度及排队类型,构建不同排队模型,分析云环境下随机请求性能.

2.2 排队模型框架

不同场景下,排队模型通常包括请求到达模式、执行模式、队列类型及排队规则等要素.基于不同场景的随机请求,请求到达模式多样.根据不同到达模式,请求到达时间间隔动态变化.当请求到达云服务系统时,如果一个请求可以产生多个任务,请求成批到达,请求成批到达其速率通常服从泊松分布[6,48-51].对请求单个到达的系统,请求到达速率分为泊松分布和一般分布;当前研究请求大多是单个到达,并且请求到达服从泊松分布[3-5,8,18,22,26-32,52];只有很少的一部分研究请求到达服从一般分布[33,53-54]和请求成批到达场景[55],请求单个到达场景通常使用在云呼叫中心和云排队叫号系统中.基于不同云服务系统,处理器处理模式多样.当处理随机请求时,同构处理器[3,18,48,56-57]服务速率相同;异构处理器[19,26,31,58-61]服务速率不同.在云服务系统中,如果处理器异构,异构处理器调度顺序影响系统性能;不同调度顺序,性能分析结果不同,处理器异构特性使

得性能分析问题更加复杂.处理器服务时间服从一般分布[36,53]和指数分布[3,27],根据处理器不同分布方式构建不同排队模型;若处理时间服从一般分布,很难确定状态转移过程进而分析系统性能,由于指数分布的无记忆性,若处理时间服从指数分布时,性能分析过程相对一般分布,比较简单.用户提出请求,请求按照排队规则随机进入云服务系统,处理器按照排队规则处理请求.排队规则受到三种因素影响:请求到达时间,用户忍耐程度和服务时间.先到先服务规则由请求到达时间确定,先到达系统的请求先接受服务.在文献[7,30,36]中,考虑用户忍耐程度,用户忍耐程度由用户最大等待时间决定.请求开始截止时间由到达时间和最大等待时间决定,开始截止时间等于请求的最大等待时间加到达时间,最早开始截止时间规则表明请求在开始截止时间点之前被处理.请求到达和处理模型关系如图 3 所示.

图 3　到达和执行模式

请求到达云系统,若处理器空闲,请求立刻得到服务;否则,基于排队规则,请求进入排队位置等待,等待处理器空闲,当某个处理器处理完当前请求,空闲处理器处理排队位置中的请求.

随机请求的性能分析问题,根据请求到达模式、执行模式和排队规则,构建单队列和多队列模型排队框架,如图 4 所示.由图 4 可知,单队列模型是多队列模型中一种特殊场景,如何基于随机请求,确定处理器调度顺序,分析和改进云环境下系统性能是一个关键问题.在单队列模型中,确定处理器个数 N,若处理器同构,处理器顺序对优化目标没有影响;若处理器异构,因为不同顺序导致性能分析优化结果不一样,还需要确定处理器顺序.针对单队列模型,根据不同场景,处理器种类和速率多种多样,请求服务时间分布不同,构建不同排队模型,对云环境下随机请求进行性能分析.在最近研究中[32],为方便分析云服务系统性能,多队列模型中单队列处理器通常同构.在图 4 中,请求以速率 λ 到达系统,每个队列有 R 个排队位置.在多队列模型中,根据朗格朗日乘子法,速率 λ 分裂到多个队列中[19],每个子队列有 R 个排队位置,单个队列中处理器同构,不同队列之间,处理器异构.S 为处理器总类型,μ_1,μ_2,…,μ_S 分别为各个队列对应处理器的处理速率,n_1,n_2,…,n_S 为每个子队列中处理器数量.在图 4 中,云提供商为服务提供商提供资源,处理器有预留实例和按需实例两种租赁模式,若预留实例能够满足系统需求,请求在预留处理器上处理;否则,租赁按需处理器处理请求.当新请求到达系统时,如果有空闲处理器,新请求分配到空闲处理器;如果系统中没有空闲处理器,当排队容量还有剩余的时候,新请求进入队列进行等待,否则请求离开系统或者系统从云服务商租赁更多处理器处理请求.针对多队列模型,每个队列中处理器通常同构,不同队列中处理器异构,如何将随机请求流分配到多个子队列,最小化请求的响应时间是多队列场景的一个关键问题.在多队列模型中,用户的最大等待时间固定或为变量.若单个队列中处理器异构,需要合理分配请求流

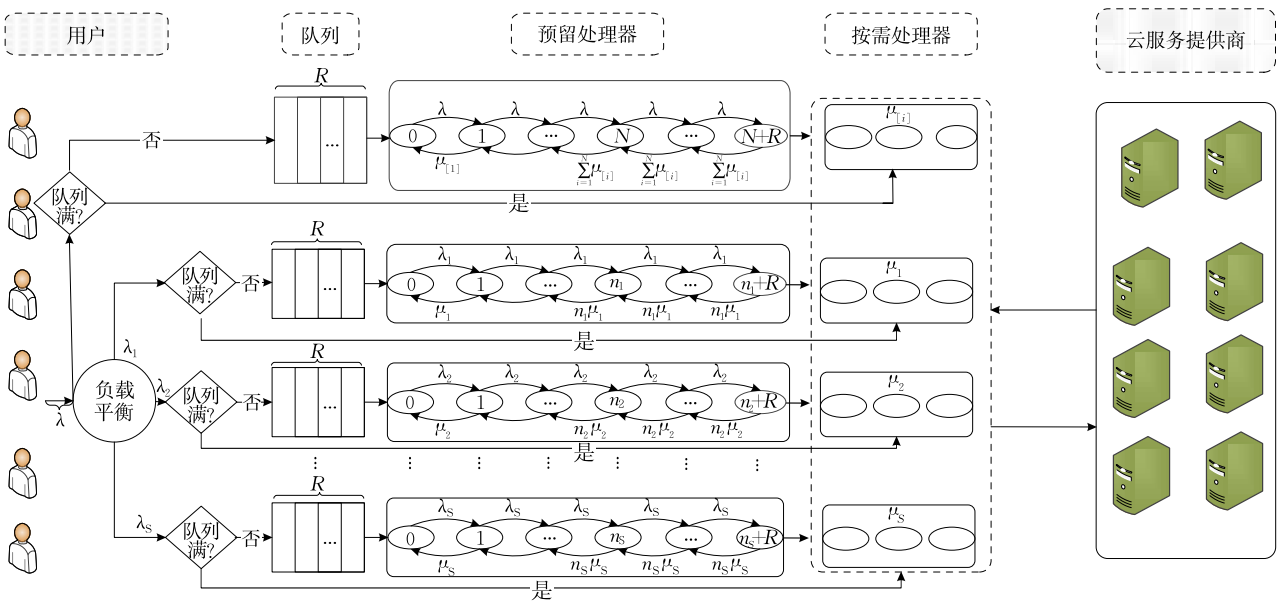

图 4　排队模型框架

和确定异构处理器顺序，因此云环境下随机请求性能分析问题更加复杂.

不同处理器数量，各种各样到达模式、执行模式和排队规则使得排队模型愈加困难，因此，基于云环境下随机请求的性能分析是很复杂的问题. 为解决这些问题，Kleinrock 采用排队论技术[10]分析云服务系统性能. 根据不同模型，现有研究采用两种方法优化目标. 在单队列模型中，对于异构处理器，选择合适处理器调度顺序优化目标. 在多队列模型中，针对多个队列采用拉格朗日乘子法平衡负载进而改进系统性能.

3 性能分析过程

动态变化的云环境下，随机请求的性能分析问题受到请求到达模式、处理器执行模式、系统队列和目标的影响. 在对随机请求分析过程中，针对不同到达模式、执行模式和队列，构建多种排队模型. 由于请求按照一定分布随机到达云服务系统，本文归纳总结四个步骤解决云计算环境下不同场景的随机请求性能分析问题.

(1) 根据系统到达模式、执行模式和队列，构建系统排队模型.

(2) 确定初始处理器类型和数量.

(3) 基于排队模型和初始处理器，定义系统状态空间，计算系统各个状态下稳态概率.

(4) 根据稳态概率，分析系统性能，调整处理器数量和类型，改进系统性能.

3.1 排队模型

基于请求到达模式、执行模式和排队规则，构建系统排队模型，确定云服务系统状态转移过程；根据给定初始状态，系统下一个状态可根据状态转移过程度量. 不同到达和执行模式使得状态转移过程不同；不同排队规则，调度请求顺序不同. 先到先服务规则通常在不考虑最大等待时间时采用；当考虑带有忍耐程度的用户时，所采用排队规则通常是最早截止时间优先.

在表 2 中比较不同到达模式、执行模式、排队规则和各种排队模型优缺点. 由表 2 可知，当到达模式 A 和执行模式 B 都是 M 时，排队模型是马尔可夫过程，容易确定系统状态转移过程，但对于云服务系统，这种假设约束条件很严格. 然而，当到达模式 A 或者执行模式 B 是 G 时，云服务系统约束条件弱化，排队过程是一个隐马尔可夫过程，很难确定转移过程. 当考虑系统中带有忍耐程度的用户时，云环境下性能分析问题变得更加复杂，但同时也更加适用于实际场景. 当 Z 是 D 或 θ 时，考虑带有忍耐程度的用户的最大等待时间固定或为变量. 云环境下随机请求性能分析问题的难易程度不同，如果请求成批到达，状态转移过程更加复杂；约束条件越松弛，云服务系统状态转移过程越复杂；结合云服务系统状态空间，确定状态转移过程，计算云服务系统稳态概率.

表 2 不同到达模式、执行模式和排队规则下比较结果

到达	A+Z	排队规则	处理器	B	文献	优点	缺点
单个	M	FCFS	同构	M	[3,27,29,32,18,52,56,57]	转移速率定	约束太强
单个	M	FCFS	同构	G	[4]	处理器约束弱化	难定转移速率
单个	M	FCFS	异构	M	[31,58-60]	转移速率确定	约束较强
单个	M	FCFS	异构	G	[5,19]	符合实际场景	难定转移速率
单个	M+D	FCFS	异构	M	[7,22,30]	考虑带有忍耐程度的用户	强约束
单个	M+D	ESF	同构	G	[35]	处理器约束弱化	难定转移速率
单个	M+D	ESF	异构	M	[17]	考虑不同的带有忍耐程度的用户	约束强
单个	G+θ	FCFS	同构	M	[54]	用户约束弱化	难定转移速率
单个	G	FCFS	同构	G	[53]	更符合实际场景	难定转移速率
成批	M	FCFS	同构	M	[6,49,51]	考虑批到达任务	强约束
成批	M	FCFS	同构	G	[28,48,55]	处理器约束弱化	难定转移速率
成批	M	FCFS	异构	M	[26,50,61]	转移速率确定	约束较强

3.2 确定处理器配置

根据不同场景，确定处理器配置的策略不同. 针对单队列模型，基于随机请求，为分析云服务性能，确定同构处理器数量和异构处理器配置及顺序很关键；针对多队列模型，还需要将请求流合理分配到多个队列. 在分析云服务系统性能时，确定处理器策略过程通常是一个迭代查找过程. 针对单队列模型，如何确定异构处理器配置和顺序是一个关键问题. 基于云环境下截止期、最大等待时间、系统容量、功耗等约束条件，确定处理器配置. 针对单队列模型，确定合适处理器顺序能够改进系统性能[62-64]. 异构处理器的数量为 N，队列容量为 R，在文献[62]中提出 r-调度策略，它表明新到达请求分配处理器概率与处理器速率有关. 令 $P(S_i)$ 是处理速率为 μ_i 时

的概率[62]，

$$p(S_i)=\frac{\mu_i^r}{\sum_{j=1}^{N}\mu_j^r} \tag{1}$$

(1) 如果 $r=0$，r-调度策略是随机处理器选取策略；

(2) 如果 $r=1$，r-调度策略是速率平衡选取策略；

(3) 如果 $r\to+\infty$，r-调度策略中处理器最快的优先被选取；

(4) 如果 $r\to-\infty$，r-调度策略中处理器最慢的优先被选取.

根据 r 的不同取值，处理器选择的策略不同，比较不同调度策略对云服务系统性能的影响.

针对异构处理器系统，采用由基尼系数 G' 构成的向量评估系统异构性[63]，根据 G' 值，随机选择处理器. 在选择合适处理器时，一般采用马尔可夫决策过程[64-65]，改进系统性能.

针对多队列排队模型，随机到达请求被分裂到多个队列中[19,66]. 对于一个多队列系统，通常采用三个步骤优化性能：

(1) 分析目标卡鲁什-库恩-塔克(KKT)[67]条件；

(2) 确定处理器数量；

(3) 确定单个队列中处理器速率.

根据目标和 KKT 约束条件，计算它们的导数，根据导数，确定处理器数量和速率，优化云服务系统性能.

3.3　概率分析

基于云环境下实际场景，分析和确定云系统的到达模式、执行模式和队列；根据到达、执行模式和队列，定义系统状态空间，确定排队模型；根据系统排队模型，确定转移速率；结合状态空间和转移速率，计算系统稳态概率. 由于排队模型的多样性，不同的排队模型，系统稳态概率不一样. 针对不同排队模型，综述不同排队模型下稳态概率计算过程.

根据不同排队模型，定义统一符号表示不同模型中处理器数量、处理器类型及排队容量. 令处理器数量是 N，不同云服务场景下，系统中的排队容量为 0、R、∞. 处理器服务速率是 $\mu_1,\mu_2,\cdots,\mu_N$，对应 S_1，$S_2,\cdots,S_N$ 处理器. 如果 $\mu_1=\mu_2=\cdots=\mu_N$，处理器同构；否则，处理器异构. 令 $\mu_{[1]},\mu_{[2]},\cdots,\mu_{[N]}$ 是一种处理器调度顺序，$\mu_{[1]}$ 首先被选择处理请求，其次为 $\mu_{[2]}$ 处理器，最后为 $\mu_{[N]}$ 处理器. 对于同构处理器，处理器顺序并不影响系统性能；然而，异构处理器顺序对系统性能影响很大. 随机请求按照先到先服务或最早开始截止时间优先准则处理任务. 为分析系统性能如拒绝率、系统中请求数量、响应时间等，要先确定处理器配置和调度顺序，结合处理器配置和调度顺序，定义云服务系统状态空间，计算系统稳态概率；基于系统稳态概率和系统状态，计算系统中请求数量、响应时间等.

3.3.1　M/B/N/N+R/FCFS 概率分析

当请求到达系统时，系统中请求数量动态变化，分析 M/B/N/N+R/FCFS 排队模型的稳态概率. 令 $\Omega=\{0,1,\cdots,N+R\}$ 为系统状态空间，每个状态表示系统中请求数量. 令 $\boldsymbol{P}=(P_0,P_1,\cdots,P_{N+R})$ 为系统处于稳态时的概率向量，P_i 代表系统状态为 i 个请求时的系统稳态概率.

本文归纳总结出 M/M/N/N+R/FCFS、M/M/N/∞/FCFS、M/M/N/N/FCFS、M/M[d]/N/N+R/FCFS，这些排队模型的统一稳态方程：

$$\begin{cases}\lambda P_0=\mu_{[1]}P_1\\ \lambda P_0+(\mu_{[1]}+\mu_{[2]})P_2=(\lambda+\mu_{[1]})P_1\\ \quad\cdots\\ \lambda P_{N-1}+\sum_{j=1}^{N}\mu_{[j]}P_{N+1}=\left(\lambda+\sum_{j=1}^{N}\mu_{[j]}\right)P_N\\ \quad\cdots\\ \lambda P_{N+R-1}=\sum_{j=1}^{N}\mu_{[j]}P_{N+R}\end{cases} \tag{2}$$

根据稳态方程，用如下无限增量矩阵 $\boldsymbol{Q}$ 表示系统转移速率：

$$\boldsymbol{Q}=\begin{pmatrix}-\lambda & \lambda & & & \\ \mu_{[1]} & -\mu_{[1]}-\lambda & \lambda & & \\ & \ddots & & & \\ & \sum_{j=1}^{N}\mu_{[j]} & -\sum_{j=1}^{N}\mu_{[j]}-\lambda & \lambda & \\ & & & \sum_{j=1}^{N}\mu_{[j]} & -\sum_{j=1}^{N}\mu_{[j]}\end{pmatrix} \tag{3}$$

根据无限增量矩阵 $\boldsymbol{Q}$，得到平衡向量方程：

$$\begin{cases}\boldsymbol{PQ}=0\\ \sum_{j=0}^{N+R}P_{j=1}\end{cases} \tag{4}$$

由方程(4)计算系统稳态概率.

3.3.2　G/G/N/∞/FCFS 概率分析

考虑请求到达时间间隔和处理时间都服从一般分布，提出一种类似于 G/G/N/∞/FCFS 排队模型分析请求响应时间和阻塞概率[53]. 任何一个一般分布都可以近似为由一系列指数分布转化而成[68]. 令

请求服务时间服从带有 c 个指数期位相型分布，任务到达时间间隔服从 e 个指数期位相型分布. 令 $\mu(k)\in(k\in\{0,1,\cdots,N+R\})$ 为当系统中有 k 个请求时处理器服务速率，$w(k)$ 是当系统中有 k 个请求时新到达请求数量. 一个 G/G/N/∞/FCFS 排队模型可转化成依赖于系统状态到达的 M/Ph/N/N+R 排队模型和依赖于状态服务的 Ph/M/N/N+R 排队模型[68]. 用 $P_k(k\in\{1,\cdots,N+R\})$ 表示请求稳态概率，根据文献[53]，稳态概率为

$$P_k=\frac{1}{G}\prod_{i=1}^{k}\frac{w(i-1)}{\mu(i)}(k\in\{0,\cdots,N+R\}) \tag{5}$$

其中，$G=1+\sum_{i=1}^{k}\prod_{j=1}^{i}\frac{w(j-1)}{w(j)}$.

3.3.3 M/M[d]/N/∞/FCFS+D 概率分析

对于带有忍耐程度的用户提出的随机请求，随机请求速率为 λ，构建 M/M[d]/N/∞/FCFS+D 排队模型[10]，分析系统稳态概率. 定义二元变量 x，当 $x>0$ 时 $h(x)=1$；其它情况，$h(x)=0$. $P_j(j\in\{1,\cdots,N+R\})$ 为当系统中有 j 个随机请求时，系统稳态概率. 根据文献[10]，经本文分析，计算系统稳态概率：

$$P_j=h(N+1-j)P_0\frac{\lambda^j}{\prod_{i=1}^{j}\sum_{k=1}^{i}\mu_{[k]}}+$$

$$h(j-N)P_0\frac{\lambda^j}{\prod_{i=1}^{N}\sum_{k=1}^{i}\mu_{[k]}\left(\sum_{k=1}^{N}\mu_{[k]}\right)^{j-N}} \tag{6}$$

为简化公式，令 $\sum=\sum_{k=1}^{N}\mu_{[k]}$ 和 $\eta=1-\frac{\lambda}{\sum}$. 如果所有处理器都在工作，新到达随机请求将在排队位置中等待，随机请求到达系统等待概率为

$$P_w=\sum_{j=N}^{\infty}P_j=\frac{P_N}{\eta} \tag{7}$$

稳态概率满足：

$$\sum_{j=0}^{\infty}P_j=1 \tag{8}$$

根据式(6)和(8)，计算 P_0：

$$P_0=\frac{1}{\sum_{j=0}^{N-1}\frac{\lambda^j}{\prod_{i=0}^{j}\left(\sum_{k=1}^{i}\mu_{[k]}\right)+\prod_{j=1}^{N}\left(\sum_{i=1}^{j}\mu_{[i]}\right)\eta}} \tag{9}$$

令请求等待时间为 w，计算 w 概率分布

$$f_w(t)=(1-P_N)v(t)+\sum P_N e^{-\eta\sum t} \tag{10}$$

其中，$v(t)$ 是单位脉冲函数[7]，定义为

$$v(t)=h\left(t-\frac{1}{z}\right)z+h\left(\frac{1}{z}-t\right).$$

另外，$v(t)$ 满足 $\int_0^{\infty}v(t)\mathrm{d}t=1$.

请求的累计概率密度函数[7]为

$$F_w(t)=1-\frac{P_N}{\eta}e^{-\eta t} \tag{11}$$

3.3.4 M[d]/M[d]/N/N+R/ESF+d 概率分析

在实际场景中，不同用户的忍耐程度不一样，请求到达速率动态变化，请求处理速率也动态变化，构建 M[d]/M[d]/N/N+R/ESF+d 排队模型，分析稳态概率. M[d]/M[d]/N/N+R/ESF+d 排队模型如图 5 所示[17]. 系统中请求由控制器从队列中调度到处理器，控制器确定请求调度顺序. 根据请求最大等待时间，请求可分 K 类. 请求最早开始截止时间由请求到达时间和最大等待时间确定，根据最早开始截止时间优先的原则，具有较高优先级请求最先被执行. 当请求实际等待时间小于最大等待时间时，请求进入队列等待. 控制器根据请求优先级将请求调度到空闲处理器.

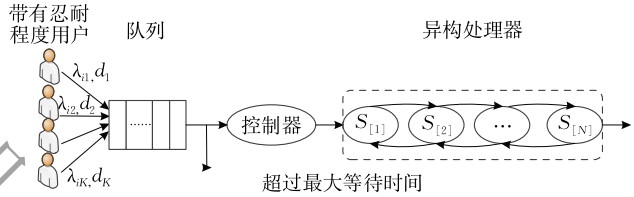

图 5 排队模型框架

定义不同优先级请求流，计算请求实际等待时间，确定系统平衡方程，计算稳态概率. 云服务提供商提供 S 种处理器，处理器的处理速率独立且服从指数分布. 令处理器处理速率为 $\mu_1,\mu_2,\cdots,\mu_S$ 且满足 $\mu_1\leq\mu_2\leq\cdots\leq\mu_S$. 服务提供商根据按需租赁模式从云提供商租赁资源. 租赁处理器数量 N 根据请求动态到达而改变. λ_i 表示系统中有 i 个请求时，即将到达请求的到达速率，λ_i 由 K 种类型的请求流 $\{\lambda_{i1},\lambda_{i2},\cdots,\lambda_{iK}\}(i\in\{0,\cdots,N+R-1\})$ 构成. K 种请求流对应最大等待时间为 $d_1,d_2,\cdots,d_K$，且满足 $d_1\leq d_2\leq\cdots\leq d_K$. 根据最大等待时间和请求到达时间，确定请求优先级. 令 $W_{(i+1)k}$ 为当系统中有 i 个请求时，第 $i+1$ 个请求为第 k 类请求的等待时间(即目标请求). 令 $\mu_{[i]}(i\in\{1,\cdots,N\})$ 为从 S 种处理器中选择 N 个处理器时，第 i 个被选择的处理器. 当系统中请求有 i 个时，如果 $i<N$，新到达的请求会被立刻处理，也就是说 $W_{(i+1)k}=0$. 如果队列中有请求在等待，表明租赁处理器都处于工作状态. $W_{(N+1)k}$ 表示当系统中一个新请求在等待时，请求等待时间与当前系统执行一个请求的剩余服务时间相同. 根据文献[10]，$W_{(N+1)k}$ 计算为

$$W_{(N+1)k}=\sum_{j=1}^{K}\frac{2\lambda_{ij}}{\left(\sum_{p=1}^{N}\mu_{[p]}\right)^2} \tag{12}$$

令 $N_{j,k}^i$ 和 $M_{j,k}^i$ 为第 j 类请求比目标请求先到达系统或者后到达系统，但却比目标请求优先被执行的数量. 当系统中有 i 个请求时，由文献[17]可知，新请求是第 k 类请求的等待时间为

$$W_{(i+1)k}=h(i-N)W_{Nk}+h(i-N-1)\frac{N_{j,k}^i+M_{j,k}^i}{\sum_{p=1}^{N}\mu_{[p]}} \tag{13}$$

将请求等待时间和最大等待时间相比较，得到系统平衡方程，根据系统的平衡方程计算各个状态稳态概率.

3.3.5　M/G/1/∞/FCFS 概率分析

在多队列模型中，如果每个子队列中只有一个处理器，并且处理时间服从一般分布，可构建 M/G/1/∞/FCFS 排队模型[69]，分析稳态概率. 令 X_n^i 为第 i 个队列中，第 n 个请求离开系统时请求数量. 令 T_n^i 为第 i 个队列中，第 n 个请求离开系统时第 $n+1$ 个请求处理时间. 令 $G(t)^i$ 为第 i 个队列中 T_n^i 的一般分布. 令 Y_n^i 为第 i 个队列中第 n 个请求离开系统时新到达系统的请求数量. 因此，$T_n^i+t_n^i$ 为第 i 个队列中第 $n+1$ 个请求离开系统的时间. 用 α_j^i 表示第 i 个队列中在第 $n+1$ 个请求被处理期间有 j 个请求到达系统的概率. X_n^i 和 Y_n^i 的关系如下[69]：

$$X_{n+1}^i=h(1-X_n^i)Y_n^i+h(X_n^i)(X_n^i+Y_n^i-1) \tag{14}$$

因此，当系统中第 $n+1$ 个请求离开系统时，当前状态请求数量仅依赖于上一个状态. $\{X_n^i\}(n\in\{1,2,\cdots\},i\in\{1,\cdots,S\})$ 构成一个内嵌马尔可夫链，系统一步转移概率如图 6[69] 所示. 根据文献[69]，α_j^i 计算为

$$\alpha_j^i=P(Y_n^i=j)=\int_0^{\infty}P(Y_n^i=j\mid T_n=t)\mathrm{d}G^i(t) \tag{15}$$

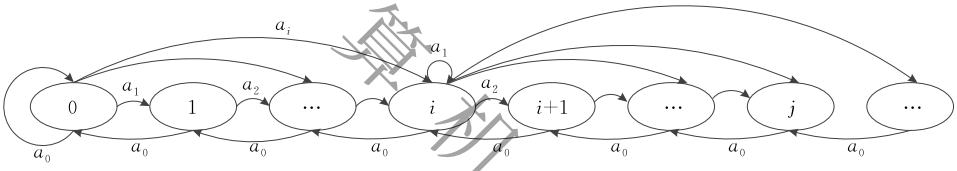

图 6　一步状态转移

令 $P(Y_n^i=j\mid T_n^i=t)$ 为处理第 $n+1$ 个请求期间，有 j 个请求到达系统的概率，λ_i 为第 i 个队列的请求到达速率. 由于请求泊松到达，$P(Y_n^i=j\mid T_n^i=t)$ 为[69]

$$P(Y_n^i=j\mid T_n^i=t)=\frac{(\lambda_i t)^j}{j!}e^{-\lambda_i t} \tag{16}$$

根据式(16)，α_j^i[69] 计算为

$$\alpha_j^i=\int_0^{\infty}\frac{(\lambda_i t)^j}{j!}e^{-\lambda_i t}\mathrm{d}G^i(t) \tag{17}$$

根据文献[70]，计算稳态概率，且 $P_j^i=\alpha_j^i$[69].

3.3.6　性能分析

经过分析各种排队模型，得到系统稳态概率，计算云服务性能指标如：系统中请求平均个数、拒绝率 P_R 和请求平均响应时间 T_r.

$$\begin{cases}L=\sum_{j=0}^{N+R}P_j j\\ P_R=P_{N+R}\\ T_r=\dfrac{L}{\lambda(1-P_R)}\end{cases} \tag{18}$$

采用以下实例，计算各个排队模型性能指标，横坐标为请求到达速率，纵坐标对应性能指标值.

(1) 针对 M/M/N/N/FCFS、M/M/N/N+R/FCFS、M/M/N/∞/FCFS 排队模型，$\lambda\in\{1,2,\cdots,20\}$，$N=7$，$\mu=4$，$R=5$，排队规则为 FCFS.

(2) 针对 M/M[d]/N/N+R/FCFS 排队模型，处理器异构，其它与以上实例参数相同，异构处理器速率为 $\mu\in\{1,2,3,4,5,6,7\}$.

(3) 针对 G/G/N/∞/FCFS 排队模型，在各个状态下，到达速率和处理速率动态变化，均值为 $\lambda\in\{1,2,\cdots,20\}$、$\mu\in\{1,2,3,4,5,6,7\}$，处理器数量及排队规则与以上相同.

(4) 针对 M/M[d]/N/∞/FCFS+D 排队模型，到达速率、处理速率及处理器数量与以上实例参数相同，排队规则为 FCFS，D 为 1.

(5) 针对 M[d]/M[d]/N/N+R/ESF+d 排队模型，处理器数量和排队位置与以上实例参数相同，任务到达速率动态变化，排队规则为 ESF.

(6) 针对 M/G/1/∞/FCFS 排队模型，只有一个处理器，任务到达速率与以上实例相同，处理器处理速率为 28，排队规则为 FCFS.

针对不同排队模型，在统一数据场景下，系统性能指标(系统中请求数量、拒绝率和响应时间)如图 7

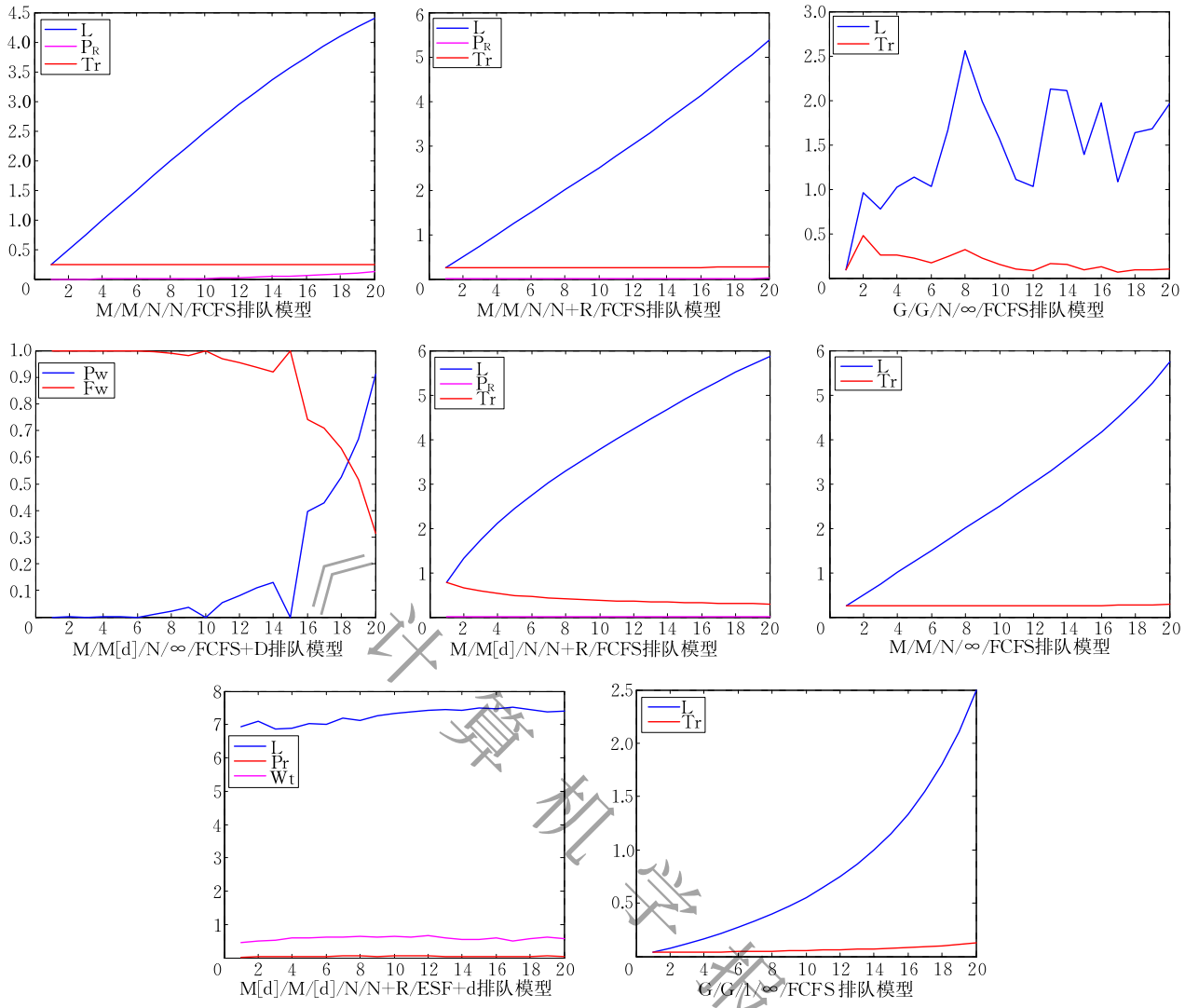

图 7 不同排队模型

所示. 由图 7 可知，随着 λ 的增加，M/M/N/N/FCFS、M/M/N/N＋R/FCFS、M/M/N/∞/FCFS、M/M[d]/N/N＋R/FCFS、M[d]/M[d]/N/N＋R/ESF＋d 和 M/G/1/∞/FCFS 排队系统中请求数量越来越多. 由于 G/G/N/∞/FCFS 排队模型中，请求到达速率和处理速率动态变化，系统中请求数量动态变化，但趋势增长. 根据 M/M[d]/N/∞/FCFS＋D 排队模型可知，由于处理器数量和速率一定，随着 λ 的增加，系统拒绝率增加.

根据不同性能指标，评价不同的优化目标. 如果系统中拒绝率过高，很多请求不能被及时处理，服务提供商收益降低. 拒绝率过高导致用户满意度降低，需要租赁更多的处理器处理请求，导致成本增加，处理器数量增加，能耗也增加. 系统中的请求数量和拒绝率决定响应时间，响应时间影响服务提供商收益，响应时间越小，单位时间内处理的请求越多，服务提供商收益越大. 在实际场景中，根据实际情况，结合不同研究目标，选择合适排队模型，结合现有一种或多种排队模型分析系统性能.

4 研究目标

不同角色关注目标不同. 服务提供商比较关注成本和收益，他们通过最小化成本和最大化收益赚取更多钱，通过最小化响应时间提高用户满意度. 云中心消耗很多能量，但是使用率却很低，因此，减少能耗对云服务系统很关键. 根据最大等待时间、队列、处理器和租赁类型，表 3 比较不同目标(成本、收益、能耗和响应时间)之间差异.

表 3　不同优化目标比较结果

目标	请求	MWT	队列	处理器	参考文献
最大化收益	泊松分布	常数	单队列	同构 & 指数分布	[7,22,30,46]
性能和能耗	泊松分布	×	单队列	异构 & 指数分布	[19,31]
性能分析	一般分布	×	单队列	同构 & 一般分布	[4,6,18,36,41,53,71]
能耗	泊松分布	×	多队列	异构 & 一般分布	[5]
性能和成本	泊松分布	常数	单队列	同构 & 指数分布	[72]
成本	泊松分布	常数	单队列	同构 & 指数分布	[17,73]

由表 3 可知，针对不同优化目标，在现有的研究中，请求大部分服从泊松分布；当考虑最大等待时间时，最大等待时间通常为常数；大部分文献考虑单队列模型，很少文献考虑多队列模型；处理器分为同构、异构，处理模式通常为指数分布. 在性能分析过程中，不同排队模型，性能分析过程相似；根据不同需求，性能分析目标不同. 基于性能分析过程，评价系统性能指标，进而预测并优化问题目标. 基于不同的排队模型，根据不同的目标，分析云环境下随机请求性能分析问题的研究现状.

4.1　成本

成本最小化已在很多现有文献中被研究. Aminizadeh 等人[20]提出一种调度方法优化动态的云资源，最小化成本. 为最小化周期性工作流的应用成本，Chen 等人[21]提出一种整数规划模型. Malawski 等人[44-45]研究最小化混合云计算场景下应用成本. Wang 等人[17]研究云计算场景中带有忍耐程度的用户的成本最小化问题. 为最小化成本和延迟，Grozev 等人[74]提出动态资源供给和负载平衡算法. 成本最小化本质是为请求提供合适的处理器进而最小化租赁和运行成本. 不同处理器选择，导致系统性能不同. 研究目标为成本最小化问题时，基于请求截止期[21]、用户最大等待时间[17]等约束，选择合适按需实例处理器，将请求合理调度到处理器上，分析并改善系统性能.

文献[72-73]采用马尔可夫模型，分析系统成本. Jagannatha 等人[73]根据马尔可夫模型和处理器分配矩阵研究成本. 在资源分配的过程中，考虑处理器故障是云计算环境中一个新方向，基于实际场景，分析处理器故障速率，确定系统状态空间，分析系统可靠性，根据马尔可夫模型预测处理器故障情况. 针对带有忍耐程度的用户，最小化操作、资源供给和性能成本[72]. 系统中有 N 个处理器，处理速率均为 μ，请求处理时间服从指数分布. 请求到达速率 λ 服从泊松分布. 令 λ_b 是故障率、λ_r 是损失率、λ_R 是系统容量为 R 时的阻塞率. P_R 为系统容量满时的拒绝率，令 C_1 是处理器单位时间内供给价格，C_2 是功耗单价，C_3 是预备排队空间单价，C_4 是带有忍耐程度的用户系统损失单价，C_5 是启动处理器单价，C_6 是系统拒绝惩罚代价，C_7 是系统中有请求时的单价，C_8 是请求在系统中的等待成本.

成本最小化问题[72]为

$$C=(NC_1+\mu C_2)\frac{\lambda}{N\mu}+RC_3+(\lambda_r+\lambda_b+\lambda_R)C_4+\lambda_r C_5+P_R C_6+L_q C_7+W_q^* C_8 \tag{19}$$

$$\text{s.t.}\quad 0\leq P_l\leq x \tag{20}$$

其中 P_l 是损失概率，W_q 是等待时间，x 是损失概率的上确界. 在文献[73]中，分析系统性能指标，损失概率 P_l 和等待时间 W_q，最小化系统成本.

文献[17]考虑带有忍耐程度的用户最大等待时间. 对于给定系统容量 ξ，请求在按需租赁处理器上被处理. 令 $\boldsymbol{n}$ 是一个 S 维向量，其中 n_i 表示第 i 种类型处理器数量，$\boldsymbol{n}$ 等价于 $\sum_{i=1}^{S} n_i$. 令 $p_i(i\in\{1,\cdots,S\})$ 对应每种类型处理器单价. 基于用户的忍耐程度，服务提供商成本最小化问题[17]为

$$\min \sum_{i=1}^{S} n_i p_i \tag{21}$$

$$\text{s.t.}\ \sum_{i=1}^{S} n_i\mu_i\geq E_\lambda \tag{22}$$

$$p_i>0 \tag{23}$$

$$P_R\leq 1-\xi \tag{24}$$

$$n_i\geq 0 \tag{25}$$

其中，E_λ 为平均到达速率，P_R 为拒绝率. 在文献[17]中，当满足系统容量 ξ 约束时，经过分析性能指标平均到达速率 E_λ 和系统拒绝率 P_R，最小化按需处理器租赁成本.

在云计算实际场景中通过最小化成本(租赁成本、操作成本等)，云服务提供商可以赚更多钱，节省更多资源，成本最小化对服务提供商意义重大. 根据不同约束条件(截止期、最大等待时间、处理器资源等)，基于请求到达模式和处理模式，构建对应排队模型，定义系统状态空间，分析和预测系统性能，进而通过选取合适处理器和租赁方式，最小化成本. 不同场景下，成本最小化问题优化指标和约束条件不

同,基于随机请求的成本最小化是一个复杂的问题,因此,云环境下基于随机请求的成本最小化性能分析问题值得深入研究.

4.2 收益

基于动态云计算环境,在文献[7,22,30,46]中,基于随机请求,分析到达模式和处理器执行模式,构建排队模型,分析系统性能,研究服务提供商收益最大化.在文献[7]中,通过租赁预留模式处理器,构建 M/M/m 排队模型使得收益最大化;确定最大等待时间的收益最大化问题.如果请求等待时间小于 D,奖励为 a.如果请求等待时间大于 D,却小于$(1+d/a)D$,惩罚为 d.根据最大等待时间约束,服务费用函数[7]为

$$C(\mu_i,w)=\begin{cases}a, & 0\leq w\leq D\\ a-d(w-D), & D<w\leq\left(1+\dfrac{d}{a}\right)D\\ 0, & w>\left(1+\dfrac{d}{a}\right)D\end{cases}\tag{26}$$

c_i是处理速率为 μ_i对应处理器的租赁成本,γ 是能耗成本.最优函数[7]为

$$G=\lambda C(\mu_i,w)-c_iN-\gamma NP_i\tag{27}$$

在文献[22]中,分析用户满意度和收益之间的关系.针对一个服务提供商,整体用户满意度是 S_a.最优函数表示[22]为

$$G_1=\lambda S_aC_1(\mu_i,w)-c_iN-\gamma NP_i\tag{28}$$

在文献[30]中,为优化配置,构建 M/M[d]/∞/FCFS+D 排队模型,最大化收益.针对异构需求用户,基于拍卖理论,建立云网络模型,最大化收益[75].在文献[76]中,考虑处理器中心,基于服务水平协议分配资源,最大化收益.针对最大化收益问题,基于随机请求,选择合适处理器,构建对应排队模型,分析并改进系统性能.当处理器一定时,请求被处理越多,收益越高.

针对服务提供商,收益是一个重要的因素,云服务提供商迫切需要最大化收益.在文献[7,22,30]中,基于随机请求,构建 M/M/N/∞/FCFS+D 系统排队模型,最大化收益.

为保证所有请求在最大等待时间内完成,通过长期和短期两种模式租赁处理器[30].如果请求等待时间大于 D,请求在短期租赁处理器上执行,否则,请求被长期租赁处理器处理.对于速率为 μ_i的处理器来说,长期租赁价格为 c_i,短期租赁价格为 c_i^s且 $c_i^s>c_i$.文献[30]中的费用函数与文献[7]中的费用函数相似,费用函数表示为

$$C_1(\mu_i,w)=\begin{cases}a, & 0\leq w\leq D\\ 0, & w>D\end{cases}\tag{29}$$

令 P_{ex}为请求超过最大等待时间概率,根据文献[77],计算 P_{ex}为

$$P_{ex}=\frac{\left(1-\dfrac{\lambda}{N\mu_i}\right)(1-P_D)}{1-\dfrac{\lambda}{N\mu_i}(1-P_D)}\tag{30}$$

因此,长期租赁成本为 $C_{long}=c_iN+\gamma NP_i$,短期租赁成本为 $C_{short}=c_i^s\dfrac{\lambda P_{ex}}{\mu_i}+\gamma\dfrac{\lambda P_{ex}}{\mu_i}P_i$.收益最大化的最优函数[30]为

$$\begin{aligned}G_2&=Revenue-C_{long}-C_{short}\\&=\lambda C_1(\mu_i,w)-c_iN-\gamma NP_i-\\&\quad c_i^s\frac{\lambda P_{ex}}{\mu_i}-\gamma\frac{\lambda P_{ex}}{\mu_i}P_i\end{aligned}\tag{31}$$

根据文献[7,22,30],收益受到租赁成本、奖励和惩罚等因素影响,请求到达模式和处理器处理模式相同,考虑不同约束下,基于性能分析的收益最大化问题,式(27)、(28)、(31)分别比较不同优化函数.将文献[22]与文献[7]相比,文献[22]考虑用户满意度;将文献[30]与文献[7]}相比,文献[30]考虑惩罚成本.如果请求不能在最大等待时间内完成,服务提供商将受到一定的惩罚.这在一定程度上,迫使服务提供商提供较好服务.在云计算中,更多收益表明服务更加高效,但是需要对应合理奖励值,因为奖励值太高,用户负担不起,奖励值太低,服务提供商不挣钱,将不会提供更好的服务.因此,为最大化收益,不同性能指标需要对应合理奖励函数,并分析系统性能.

4.3 响应时间

请求等待时间取决于调度策略,由系统中等待请求数量和处理器配置影响,处理时间取决于处理器配置.请求响应时间等于请求等待时间加上处理时间,是性能分析中的一个重要指标.在文献[5]中,研究一种启发式解析解方法,最小化带有功率约束任务响应时间.在文献[19],平衡请求响应时间和功耗.在文献[31]中,通过分析不同速率模式最小化请求响应时间.文献[19,31]请求到达模式相同,处理器处理模式不同.基于服从一般分布的到达和执行模式,计算请求响应时间、系统中请求数量、使用率和请求吞吐量[53].文献[19,31]请求到达模式不同,处理器处理模式相同.Nguyen 等人[78]构建三状态模型,减少请求等待时间.Bharkad 等人[79]分析有限处理器系统动态行为,通过增加处理器的方式降

低请求等待时间.基于随机请求,分析请求响应时间是性能分析的关键问题.

在文献[31]中,研究负载依赖于动态功率管理技术,最小化响应时间.基于随机请求,构建 M/M/N/∞/FCFS 排队模型,计算请求响应时间和系统中请求数量.基于请求负载,为处理器分配合适功率;基于给定功率约束,调节处理器速率模式,减少请求响应时间;不同的请求负载,系统提供功率不一样.在文献[31]中,提出两种速率模型:空闲速率模型和常速率模型.根据系统中请求数量和功耗约束条件,使用朗格朗日乘子法选择合适处理器,最小化响应时间.

对于一个多优先级抢占的 M/G/1 排队系统,根据最早截止期优先准则,构建一种排队理论分析模型[35],分析请求等待时间.最早截止期优先准则平衡各个不同优先级请求,截止期越早,优先级越高;高优先级任务优先被执行,截止期早的请求也能被服务,增加系统吞吐量.第 i 个级别请求等待时间[35]为

$$W_i=W_0^i+\sum_{k=1}^{i}\rho_k W_k+\sum_{k=i+1}^{K}\rho_k\max(0,W_k-D_{k,i})+\sum_{k=1}^{i-1}\rho_k\max(W_i,D_{i,k}) \tag{32}$$

其中,W_0^i 是第 i 个优先级请求在系统中平均剩余时间;$D_{k,i}$ 是第 i 个优先级请求和第 k 个优先级请求截止期差值;ρ_k 是第 k 个优先级请求负载.由处理器速率确定请求处理时间,计算请求响应时间.

在文献[31,36]中,请求到达模式相同,处理器处理模式不同.由于存在带有忍耐程度的用户,若响应时间小于用户最大等待时间,用户满意度提高,否则,用户满意度下降,因此最小化响应时间可以提高用户满意度.为分析随机请求响应时间,根据转移速率确定系统中请求数量;通过最小化请求响应时间,在一个固定时间段,系统处理更多请求;因此,最小化响应时间在性能分析过程中很重要.

4.4　能耗

最小化能耗对云提供商至关重要,基于随机请求分析系统能耗是云计算环境中一个关键问题.据估计,云中心仅 2014 年消耗 700 亿千瓦时的电力,约占美国总电力消耗的 1.8%[80].从 2010 年到 2014 年,云中心用电量增长约 4%,从 2014 年到 2020 年之间的能源使用量仍将增长 4%,根据目前趋势估计,到 2020 年,美国云计算中心预计将消耗约 730 亿千瓦时电量[80].根据国际能源署的新政策方案,该方案考虑现有和计划中政府政策规定,世界能源需求从 2012 年到 2040 年预计将增长 37%[81].能耗已被许多研究人员研究,在现有研究中,基于能耗问题,通常有 3 类:

(1)在性能满足一定条件时,最小化能耗;

(2)给定功率约束,优化系统性能;

(3)平衡能耗和性能.

在基于随机请求性能分析过程中,能耗与处理器功率和请求响应时间密切相关.Mitrani 建立排队模型满足高性能和低功耗[46-47].Zheng 等人[82]构建带有约束马尔可夫决策模型为网络处理器集群提供功率管理.通过为处理器分配功率,Li[5]优化数据中心中处理器整体服务质量.

在文献[31]中,给定功率,通过分析不同速度模式最小化请求响应时间.在文献[83]中,Chen 等人为移动设备构建半马尔可夫决策过程模型,在应用程序执行时间和功耗之间达到良好平衡.Qiu 等人[71]采用半马尔可夫模型,Laplace-Stieltjes 变换(LST)贝叶斯方法,通过一种使用重试故障恢复机制分析云服务可靠性-性能(R-P)和可靠性-能量(R-E).Entezari-Maleki 等人[84]构建一个随机活动网络模型评估云计算中处理器功耗和性能.Zhou 等人[85]提出两种新颖的自适应能量感知算法,最大化云数据中心能耗和最小化服务水平协议违反率.Sayadnavard 等人[86]提出一种新方法平衡可靠性和能量效率.

根据文献[31],处理器功率消耗由 $P=\omega CV^2\eta$ 确定.其中,ω 是转化量、C 是电容量、V 是电压、η 是时钟频率.对于任意一个速率为 μ 的处理器,满足 $\mu\propto\eta$ 和 $\eta\propto V^{\phi}$ 且 $0<\phi\leq 1$.$\eta\propto V^{\phi}$ 表明 $V\propto\eta^{\frac{1}{\phi}}$.根据文献[25],$\mu\propto\eta$ 和 $V\propto\eta$ 代表 $P\propto\mu^{\alpha}$.其中 $\alpha=1+2/\varphi\geq 3$,即 P 表示为 $\kappa\mu^{\alpha}$,κ 是常数[25].

$$P=\kappa\mu^{\alpha}+P^{*} \tag{33}$$

其中,P^{*} 是静态消耗功率.

为提高系统性能和降低能耗,在文献[31]中采用负载依赖动态功率管理技术,分析两速率模式 (b,s_1,s_2).令 r 为请求指令要求,服从均值为 $\bar{r}$ 的指数分布.处理器服务速率为 $\frac{s_i}{\bar{r}}(i\in\{1,2\})$.每种类型处理器功率消耗为 κs_1^{α}.两速率模式优化问题考虑参数有 $N,\lambda,P,b,\bar{r}$.令 $P_i(i\in\{0,1,\cdots,+\infty\})$ 为系统中有 i 个请求时稳态概率,基于两速率模式,空闲速率模式下平均功率消耗[31]为

$$P(s_1,s_2)=$$

$$P_0\left(\left(\sum_{i=1}^{N-1}\frac{\lambda^i}{\mu_1^i i!}+\frac{N^{N+1}\left(\left(\frac{\lambda}{N\mu_1}\right)^N-\left(\frac{\lambda}{N\mu_1}\right)^b\right)}{1-\frac{\lambda}{N\mu_1}}\right)\kappa s_1^\alpha+\frac{N^{N+1}}{N!}\left(\frac{\lambda}{N\mu_1}\right)^b\frac{\frac{\lambda}{N\mu_2}}{1-\frac{\lambda}{N\mu_2}}\kappa s_2^\alpha\right)NP^* \tag{34}$$

常速率模式下功率消耗[31]为

$$P(s_1,s_2)=$$

$$P_0\left(\left(\sum_{i=1}^{N-1}\frac{\lambda^i}{\mu_1^i i!}+\frac{N^{N+1}\left(\left(\frac{\lambda}{N\mu_1}\right)^N-\left(\frac{\lambda}{N\mu_1}\right)^b\right)}{1-\frac{\lambda}{N\mu_1}}\right)NP_0\kappa s_1^\alpha+\frac{N^{N+1}}{N!}\left(\frac{\lambda}{N\mu_1}\right)^b\frac{\frac{\lambda}{N\mu_2}}{1-\frac{\lambda}{N\mu_2}}\kappa s_2^\alpha\right)NP^* \tag{35}$$

根据能耗约束，功率被最优分配到异构处理器上，最小化请求响应时间[5]. 一个空闲处理器消耗功率为静态功率 P^*，U_i 为处理器 i 的使用率，给定功率 P_i，处理器 i 在 T 时间内消耗能量[5]为

$$E_i=T(P_iU_i+P^*) \tag{36}$$

总能量消耗为

$$E=\sum_{i=1}^{N}E_i \tag{37}$$

随着请求动态变化，功率动态变化导致处理速率变化，处理速率变化导致请求响应时间变化，在基于随机请求性能分析过程中，能耗随着功耗和请求响应时间变化.

5 比较和展望

本文对云环境下随机到达请求的性能分析进行综述，采用符号表示方法 A/B/C/C+X/Y+Z 对排队系统进行建模，其中包括到达模式、执行模式、处理器数量、最大队列容量、队列规则和用户的忍耐程度. 根据排队框架比较不同的排队模型，并根据不同排队模型分析各种不同目标，采用不同方法分析系统性能. 对单队列排队模型，确定各种处理器顺序是一个关键问题. 基于请求的随机性，采用马尔可夫决策过程确定处理器顺序，分析系统性能. 对于多队列排队模型，如何将总到达速率分到不同队列以平衡工作负载是一个关键问题，通常采用拉格朗日乘子法拆分到达速率使得负载平衡. 对于多队列排队模型中任意一个队列，它类似于单队列模型. 针对现有排队模型进行比较，分析不同排队模型适用性和局限性，展望云环境下随机请求性能分析未来研究方向.

5.1 比较

针对单个到达请求，请求独立，而批量到达请求是一种可产生小任务的超级请求；针对不同执行模式，当处理器异构时，比同构处理器更难分析系统性能. 不同到达和服务处理方式使问题变得不同，本文提出的排队模型框架适用于很多场景，通过该框架描述排队模型，基于随机请求，分析云服务系统性能. 如果服务时间服从一般分布，在实际情况中更常见，如何确定系统稳态概率更加复杂.

为分析系统性能，针对不同约束条件如最大等待时间、有限处理器和队列容量，构建不同排队模型. 对于单个云服务中心构成的系统，通常构建单队列模型，对于多个异地云服务中心构成的系统，通常构建多队列模型，请求随机到达系统，若处理器空闲，请求被处理器处理；否则，若队列有位置，请求在队列中等待；若队列满，请求离开系统. 不同排队模型性能分析过程相似，问题复杂度取决于请求到达时间分布和处理器服务时间分布，在不同场景下，不同模型的转移速率不一样，在不同的约束条件下，根据转移速率计算稳态概率，最后，根据稳态概率，计算系统性能指标.

在表 4 中比较各种各样的排队模型，由表 4 可知，将 M/M/N/∞/FCFS、M/M/N/N+R/FCFS 与 M/M/N/N/FCFS 相比较，系统中的排队容量不同. M/M/N/N/FCFS 没有排队位置，如果没有空闲处理器，请求将被拒绝. M/M/N/N+R/FCFS 排队容量固定，有 R 个，而 M/M/N/∞/FCFS 排队位置有无限个，如果没有空闲处理器，任务将一直在排队位置等待，直到被处理. M/G/N/N+R/FCFS 和 M/M/N/N+R/FCFS 的区别在于 M/G/N/N+R/FCFS 处理时间服从一般分布，而 M/M/N/N+R/FCFS 处理时间服从指数分布. G/G/N/N+R/FCFS 和 M/G/N/N+R/FCFS 区别在于 G/G/N/N+R/FCFS 请求到达时间间隔服从一般分布，而 M/G/N/N+R/FCFS 服从指数分布. 通过比较 M/M[d]/N/N+R/FCFS+D 和 M/M/N/N+R/FCFS、M/M[d]/N/N+R/FCFS+D 排队规则是最早开始截止时间优先，并且最大等待时间是常数. 将 M[d]/M/N/N/FCFS、M[d]/M/N/N+R/FCFS 和 M/M/N/N/FCFS、M/M/N/N+R/FCFS 相比较，请求到达时间间隔动态变化且服从指数分布，

M[d]/M/N/N/FCFS、M[d]/M/N/N+R/FCFS 请求到达的时间间隔期望动态变化，而 M/M/N/N/FCFS、M/M/N/N+R/FCFS 请求到达的时间间隔期望不变. 将 M[d]/M[d]/N/N+R/ESF+D 与 M[d]/M/N/N+R/FCFS 相比较，请求处理时间动态变化，且服从指数分布. 在排队模型 M[d]/M[d]/N/N+R/ESF+D 和 M/G/1/∞/ESF+D 中，最大等待时间是一个常数且排队规则为最早开始截止时间优先. 在排队模型 M/G/1/∞/ESF+D 中，处理时间服从一般分布，且排队容量无限. 在排队模型 G/M/N/∞/ESF+θ 中，到达模式服从一般分布并且请求最大等待时间动态变化. 根据表 4 可知，现有研究考虑各种各样单队列模型场景，只有小部分研究考虑多队列模型场景.

表 4　不同排队模型比较结果

队列	模型	文献	适用性	局限性
单队列	M/M/N/N+R/FCFS	[3,6,27,18,56,72]	到达、处理模式及排队规则确定	约束条件强
单队列	M/M/N/N/FCFS	[29,73]	到达、处理模式及排队规则确定	没有排队位置
单队列	M/G/N/N+R/FCFS	[4]	处理模式服从一般分布	到达模式约束强
单队列	G/G/N/N+R/FCFS	[53]	到达和处理模式服从一般分布	排队容量限定
单队列	M/M/N/N+R/FCFS+D	[7,22,30]	考虑带有忍耐程度的用户	约束条件强
单队列	M[d]/M/N/N/FCFS	[60]	到达模式动态变化	没有排队位置
单队列	M/M[d]/N/∞/FCFS	[31,59]	处理模式动态变化	排队容量无限
单队列	M[d]/M/N/N+R/FCFS	[58]	到达模式动态变化	排队容量限定
单队列	M[d]/M[d]/N/N+R/ESF+D	[17]	到达和处理模式动态变化	带有忍耐程度的用户
单队列	M/G/1/∞/ESF+D	[35]	处理模式服从一般分布	只有一个处理器
单队列	G/M/N/∞/ESF+θ	[54]	到达模式服从一般分布	考虑带有忍耐程度的用户
多队列	M/M/N/∞/FCFS	[41,46]	到达、处理模式及排队规则确定	约束条件强
多队列	M/G/1/∞/FCFS	[5,19]	处理模式服从一般分布	每个队列只有一个处理器

根据实际场景，当新的实际场景符合这些条件时，基于已有模型，构建新的排队模型. 在构建排队模型过程中，若处理器同构，不用确定处理器顺序；否则，若处理器异构，结合性能分析过程中优化方法确定处理器顺序和类型. 若排队模型符合表 4 中的一种或者多种场景，如在文献[87]中是 M/M/1/1+R 排队模型和 M/M/N/N+R 排队模型的组合，依据排队模型，对系统进行性能分析，依据性能分析结果，对系统进行改进和优化，具有实际指导意义.

基于随机请求，由于不同处理器具有不同价格，不同的处理器功率不同，通过选择合适处理器，将请求合理分配到处理器，优化请求响应时间、成本、收益和能耗. 这些目标相互影响，最小化成本与最大化收益相关，能量消耗最小化则降低电力成本. 若处理器的功率比较小，其处理速率较小，响应时间则增加. 响应时间增加，单位时间内处理的请求数量减少，收益减少. 这些优化目标，都要通过构建排队模型，分析系统性能后，将性能分析结果反馈到系统中，对系统参数进行调整，通过对系统参数迭代调整，确定合适处理器配置，进一步优化系统性能. 通过调整系统参数提高系统性能，在性能分析过程中优化目标.

5.2　展望

基于随机请求对云环境下系统性能分析时，不同的到达模式、执行模式和队列构造不同的排队模型. 不同的排队模型和优化目标，使得云计算环境下的性能分析问题更加复杂和多样. 当随机请求到达系统服从不同分布，及处理器处理请求服从不同分布时，性能分析问题不同. 已构建的排队模型能够解决条件相同，模型相同的问题，由于实际情况中，约束多样，不同约束之间的组合产生不同排队模型，不同排队模型性能分析过程类似，但性能指标计算过程不同. 在对云环境下随机请求进行性能分析时，仍然有很多未解决的问题. 针对云场景的复杂和多样性，从以下四个方面基于随机请求，对云环境下系统性能进行深入研究：

(1) 基于随机请求分析云环境下系统性能时，对于不同到达和执行模式，请求到达时间间隔和处理时间假设为特殊分布，根据给定场景，分析请求到达时间间隔和处理时间，根据分析结果，构建排队模型. 这种假设符合一些特殊实际场景，无法适用于某些一般场景. 一些请求到达服从一般分布场景在已有模型中还没有被研究，比如 G/M/N/∞/FCFS 排队模型和 G/D/N/∞/FCFS 排队模型，分别解决随机请求到达服从一般分布、同构处理器执行时间服从指数分布、排队位置无限和随机请求到达服从一般分布、同构处理器处理时间固定、排队位置无限的性能分析问题. 针对带有忍耐程度的用户，G[d]/G/N/N+R/ESF+D 排队模型和 G/D[d]/N/N+R/ESF+D 排队模型分别解决随机请求到达服从动态

一般分布，处理器处理时间服从一般分布、排队容量有限、带有固定最大等待时间约束和随机请求到达服从一般分布、异构处理器处理时间固定、排队容量有限、带有固定最大等待时间约束的性能分析问题.在现有研究中，我们基于请求截止期约束，根据实际场景，分析请求到达模式和处理器执行模式，选择合适处理器，构建排队模型，最小化能耗.

(2)针对批到达请求，单位时间内到达请求是常数或者变量.由于批到达速率的动态特性，系统中转移速率也将发生变化.当执行模式和队列确定，请求到达场景为批到达时，如何处理批到达随机请求性能分析是值得研究的课题.基于批到达请求，统计请求到达概率分布，及处理器处理批请求的概率分布，构建新的排对模型，分析系统性能，预测系统性能指标.

(3)针对一个多队列系统，每个队列中处理器异构，根据请求到达速率实时性，处理器数量动态变化.多队列中，每个队列中模型大多是一些特殊场景[32,40-44].在将来研究中，基于随机请求进行研究时，更多一般场景将会被研究.在多队列系统中，考虑请求最大等待时间，排队模型 M/G/N/N+R/ESF+D 和 M/D/N/N+R/ESF+D 将被研究.在正在研究的工作中，我们基于带有忍耐程度的用户，考虑多队列场景下，针对每个队列构建 M/M/N/N+R/FCFS+θ 排队模型，最大化收益.在多队列模型中，处理器处理时间服从一般分布.数据传送到处理器时，需要时间，在构建新排队模型时，考虑延迟时间.在处理随机请求过程中，处理器有时候会发生故障或者老化，正在被处理的请求要发生迁移，在故障的系统中，分析系统性能时，要考虑迁移时间.统计处理器故障概率分布，构建排队论系统，分析系统可靠性.统计处理器老化分布，分析处理器生命周期.

(4)当随机请求到达云系统时，随着处理器数量和类型增加，系统中状态空间是巨大的，这将导致无法计算状态稳态概率，因此，无法评估系统性能.由于处理随机请求时，通常采用马尔可夫决策过程，在对巨大状态空间进行处理时，结合启发式方法将问题分解成有小空间状态构成的子问题.这在以后的研究中，是一个很好的解决方案.

6 结 论

基于随机到达系统的请求，根据实际场景，分析请求到达模式、处理器执行模式；构建单队列或多队列排队模型；定义系统状态空间，确定系统转移速率；计算云服务系统不同状态的稳态概率，结合稳态概率，分析系统性能，优化云服务系统目标.系统性能目标包含成本、收益、响应时间和能耗，不同目标之间互相影响.请求到达速率和处理器处理速率根据不同分布进行分类，请求到达速率通常服从泊松分布，处理时间通常服从指数分布；针对一般场景，请求到达速率和处理时间服从一般分布；不同到达模式和执行模式下，排队模型不同，性能分析过程类似.当到达模式和执行模式服从一般分布时，很难确定系统状态转移速率；当到达模式和执行模式服从特殊分布时，容易确定系统状态转移速率，但这种特殊分布只适用于特殊场景.随机请求带有截止期或最大等待时间约束，基于不同约束条件，可采用先到先服务和最早开始截止时间优先准则.当采用不同准则时，请求调度顺序不同.针对单队列模型系统通常对单个云数据中心进行性能分析，多队列模型系统对多个数据中心构成系统采取不同方法对系统进行性能分析.本文针对不同模型，建立统一排队模型分析框架，提出性能分析通用步骤，分析和比较已有排队模型，分析系统性能，展望未来研究方向.

参 考 文 献

[1] Bermbach D, Wittern E, Tai S. Cloud Service Benchmarking: Measuring Quality of Cloud Services from a Client Perspective. Switzerland: Springer International Publishing, 2017

[2] Shuang Wang, Xiao-Ping Li, Ruiz R. Performance analysis for heterogeneous cloud servers using queueing theory. IEEE Transactions on Computers, 2020, 69(4): 563-576

[3] Ghosh R, Trivedi K S, Naik V K, Kim D S. End-to-end performability analysis for infrastructure-as-a-service cloud: An interacting stochastic models approach//Proceedings of the IEEE Pacific Rim International Symposium on Dependable Computing. Tokyo, Japan, 2010: 125-132

[4] Khazaei H, Mišić J, Mišić V B. Performance analysis of cloud computing centers using M/G/M/M + R queuing systems. IEEE Transactions on Parallel and Distributed Systems, 2011, 23(5): 936-943

[5] Ke-Qin Li. Optimal power allocation among multiple heterogeneous servers in a data center. Sustainable Computing: Informatics and Systems, 2012, 2(1): 13-22

[6] Khazaei H, Mišić J, Mišić V B, Rashwand S. Analysis of a pool management scheme for cloud computing centers. IEEE Transactions on Parallel and Distributed Systems, 2013, 24(5): 849-861

[7] Cao Jun-Wei, Hwang Kai, Li Ke-Qin, Zomaya A Y. Optimal multiserver configuration for profit maximization in cloud computing. IEEE Transactions on Parallel and Distributed Systems, 2014, 24(6): 1087-1096

[8] Zhe-Xi Yang, Wei Liu, Duo Xu. Study of cloud service queuing model based on embedding Markov chain perspective. Cluster Computing, 2018, 21(1): 837-844

[9] El Kafhali S, Salah K. Stochastic modeling and analysis of cloud computing data center//Proceedings of the 2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN). Paris, France, 2017: 122-126

[10] Kleinrock L. Queueing Systems, Volume 2: Computer Applications. New York: Wiley, 1976

[11] Benedict S. Performance issues and performance analysis tools for HPC cloud applications: A survey. Computing, 2013, 95(2): 89-108

[12] Ward A R. Asymptotic analysis of queueing systems with reneging: A survey of results for FIFO, single class models. Surveys in Operations Research and Management Science, 2012, 17(1): 1-14

[13] Balsamo S, de Nitto Personè V. A survey of product form queueing networks with blocking and their equivalences. Annals of Operations Research, 1994, 48(1): 31-61

[14] Li Jian-Qiang, Fan Yu-Shun. A method of workflow model performance analysis. Chinese Journal of Computers, 2003, 26(5): 513-523(in Chinese)
(李健强，范玉顺. 一种工作流模型的性能分析方法. 计算机学报，2003，26(5)：513-523)

[15] Ram R, Viswanadham N. Recent advances in queueing networks: A survey with applications to automated manufacturing//Proceedings of the XVI Annual Convention and Exhibition of the IEEE in India. Bangalore, India, 1991: 89-93

[16] Schwarz J A, Selinka G, Stolletz R. Performance analysis of time-dependent queueing systems: Survey and classification. Omega, 2016, 63: 170-189

[17] Shuang Wang, Xiao-Ping Li, Ruiz R, Yun Wang. Cost minimization for service providers with impatient consumers in cloud computing//Proceedings of the 2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD). Porto, Portugal, 2019: 374-379

[18] Yun-Ni Xia, Meng-Chu Zhou, Xin Luo, et al. Stochastic modeling and quality evaluation of infrastructure-as-a-service clouds. IEEE Transactions on Automation Science and Engineering, 2015, 12(1): 162-170

[19] Yuan Tian, Chuang Lin, Ke-Qin Li. Managing performance and power consumption tradeoff for multiple heterogeneous servers in cloud computing. Cluster Computing, 2014, 17(3): 943-955

[20] Aminizadeh L, Yousefi S. Cost minimization scheduling for deadline constrained applications on vehicular cloud infrastructure//Proceedings of the International Conference on Computer and Knowledge Engineering. Mashhad, Iran, 2014: 358-363

[21] Long Chen, Xiao-Ping Li, Ruiz R. Resource renting for periodical cloud workflow applications. IEEE Transactions on Services Computing, 2020, 13(1): 130-143

[22] Jing Mei, Ken-Li Li, Ke-Qin Li. Customer-satisfaction-aware optimal multiserver configuration for profit maximization in cloud computing. IEEE Transactions on Sustainable Computing, 2017, 2(1): 17-29

[23] Ataie E, Entezari-Maleki R, Rashidi L, et al. Hierarchical stochastic models for performance, availability, and power consumption analysis of IaaS clouds. IEEE Transactions on Cloud Computing, 2017, 7(4): 1039-1056

[24] Ataie E, Entezari-Maleki R, Etesami S E, et al. Power-aware performance analysis of self-adaptive resource management in IaaS clouds. Future Generation Computer Systems, 2018, 86: 134-144

[25] Bo Zhai, Blaauw D, Sylvester D, Flautner K. Theoretical and practical limits of dynamic voltage scaling//Proceedings of the 41st annual Design Automation Conference. New York, USA, 2004: 868-873

[26] Zhong-Ju Zhang, Daigle J. Analysis of job assignment with batch arrivals among heterogeneous servers. European Journal of Operational Research, 2012, 217(1): 149-161

[27] Ghosh R, Longo F, Naik V K, Trivedi K S. Modeling and performance analysis of large scale IaaS clouds. Future Generation Computer Systems, 2013, 29(5): 1216-1234

[28] Kaur P D, Chana I. A resource elasticity framework for QoS-aware execution of cloud applications. Future Generation Computer Systems, 2014, 37: 14-25

[29] Longo F, Ghosh R, Naik V K, Trivedi K S. A scalable availability model for infrastructure-as-a-service cloud//Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks. Hong Kong, China, 2011: 335-346

[30] Jing Mei, Kenli Li, Aijia Ouyang, Ke-Qin Li. A profit maximization scheme with guaranteed quality of service in cloud computing. IEEE Transactions on Computers, 2015, 64(11): 3064-3078

[31] Ke-Qin Li. Improving multicore server performance and reducing energy consumption by workload dependent dynamic power management. IEEE Transactions on Cloud Computing, 2015, 4(2): 122-137

[32] Legros B. Waiting time based routing policies to parallel queues with percentiles objectives. Operations Research Letters, 2018, 46(3): 356-361

[33] Ferragut A, Rodriguez I, Paganini F. Optimal timer-based caching policies for general arrival processes. Queueing Systems, 2018, 88(3-4): 207-241

[34] Zhang Shou-Li, Liu Chen, Han Yan-Bo, Li Xiao-Hong. DANCE: A service adaption approach for the dynamic integration of cloud and edge. Chinese Journal of Computers,

2020, 43(3): 423-439(in Chinese)
(张守利, 刘晨, 韩燕波, 李晓红. DANCE: 一种面向云-端动态集成的服务适配方法. 计算机学报, 2020, 43(3): 423-439)

[35] Hai-Yang Qian, Medhi D, Trivedi K. A hierarchical model to evaluate quality of experience of online services hosted by cloud computing//Proceedings of the International Symposium on Integrated Network Management (IM 2011) and Workshops. Dublin, Ireland, 2011: 105-112

[36] Abhaya V G, Tari Z, Zeephongsekul P, Zomaya A Y. Performance analysis of EDF scheduling in a multi-priority preemptive M/G/1 queue. IEEE Transactions on Parallel and Distributed Systems, 2013, 25(8): 2149-2158

[37] Chao Shen, Wei-Qin Tong, Jenq-Neng Hwang, Qiang Gao. Performance modeling of big data applications in the cloud centers. The Journal of Supercomputing, 2017, 73(5): 2258-2283

[38] Salah K, Sheltami T R. Performance modeling of cloud apps using message queueing as a service (MaaS)//Proceedings of the 2017 20th Conference on Innovations in Clouds, Internet and Networks. Paris, France, 2017: 65-71

[39] Movaghar A. Analysis of a dynamic assignment of impatient customers to parallel queues. Queueing Systems, 2011, 67(3): 251-273

[40] Delasay M, Kolfal B, Ingolfsson A. Maximizing throughput in finite-source parallel queue systems. European Journal of Operational Research, 2012, 217(3): 554-559

[41] Knessl C, Matkowsky B, Schuss Z, Tier C. Two parallel queues with dynamic routing. IEEE Transactions on Communications, 1986, 34(12): 1170-1175

[42] Raei H, Yazdani N, Shojaee R. Modeling and performance analysis of cloudlet in mobile cloud computing. Performance Evaluation, 2017, 107: 34-53

[43] Zhiping Peng, Delong Cui, Jinglong Zuo, et al. Random task scheduling scheme based on reinforcement learning in cloud computing. Cluster Computing, 2015, 18(4): 1595-1607

[44] Malawski M, Figiela K, Nabrzyski J. Cost minimization for computational applications on hybrid cloud infrastructures. Future Generation Computer Systems, 2013, 29(7): 1786-1794

[45] Kenli Li, Chubo Liu, Keqin Li, Zomaya A Y. A framework of price bidding configurations for resource usage in cloud computing. IEEE Transactions on Parallel and Distributed Systems, 2015, 27(8): 2168-2181

[46] Mitrani I. Managing performance and power consumption in a server farm. Annals of Operations Research, 2013, 202(1): 121-134

[47] Khazaei H, Mišić J, Mišić V B. Performance of cloud centers with high degree of virtualization under batch task arrivals. IEEE Transactions on Parallel and Distributed Systems, 2012, 24(12): 2429-2438

[48] Wei Li, Fretwell R J, Kouvatsos D D. Performance analysis of queues with batch Poisson arrival and service//Proceedings of the 2011 IEEE 13th International Conference on Communication Technology. Jinan, China, 2011: 1033-1036

[49] Shorgin S, Pechinkin A, Samouylov K, et al. Queuing systems with multiple queues and batch arrivals for cloud computing system performance analysis//Proceedings of the 2014 International Science and Technology Conference (Modern Networking Technologies) (MoNeTeC). Moscow, Russia, 2014: 1-4

[50] Khazaei H, Mišić J, Mišić V B. A fine-grained performance model of cloud computing centers. IEEE Transactions on Parallel and Distributed Systems, 2012, 24(11): 2138-2147

[51] Kowsigan M, Balasubramanie P. An efficient performance evaluation model for the resource clusters in cloud environment using continuous time Markov chain and Poisson process. Cluster Computing, 2019, 22(5): 12411-12419

[52] Atmaca T, Begin T, Brandwajn A, Castel-Taleb H. Performance evaluation of cloud computing centers with general arrivals and service. IEEE Transactions on Parallel and Distributed Systems, 2015, 27(8): 2341-2348

[53] Bruneel H, Maertens T. A discrete-time queue with customers with geometric deadlines. Performance Evaluation, 2015, 85: 52-70

[54] Henderson W, Taylor P G. Product form in networks of queues with batch arrivals and batch services. Queueing Systems, 1990, 6(1): 71-87

[55] Bruneo D. A stochastic model to investigate data center performance and QoS in IaaS cloud computing systems. IEEE Transactions on Parallel and Distributed Systems, 2013, 25(3): 560-569

[56] Xiwei Qiu, Yuanshun Dai, Yanping Xiang, Liudong Xing. A hierarchical correlation model for evaluating reliability, performance, and power consumption of a cloud service. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2015, 46(3): 401-412

[57] Misra C, Swain P K. Performance analysis of finite buffer queueing system with multiple heterogeneous servers//Proceedings of the International Conference on Distributed Computing and Internet Technology. Springer, Berlin, Heidelberg, 2010: 180-183

[58] Alves F S Q, Yehia H C, Pedrosa L A C, et al. Upper bounds on performance measures of heterogeneous M/M/c queues. Mathematical Problems in Engineering, 2011, Article ID 702834

[59] Tirdad A, Grassmann W K, Tavakoli J. Optimal policies of M(t)/M/c/c queues with two different levels of servers. European Journal of Operational Research, 2016, 249(3): 1124-1130

[60] Khazaei H, Mišić J, Mišić V B, Mohammadi N B. Modeling the performance of heterogeneous IaaS cloud centers//Proceedings of the 2013 IEEE 33rd International Conference

on Distributed Computing Systems Workshops. Philadelphia, USA, 2013: 232-237

[61] Doroudi S, Gopalakrishnan R, Wierman A. Dispatching to incentivize fast service in multi-server queues. ACM SIGMETRICS Performance Evaluation Review, 2011, 39(3): 43-45

[62] Li Na, Stanford D A. Multi-server accumulating priority queues with heterogeneous servers. European Journal of Operational Research, 2016, 252(3): 866-878

[63] Rykov V, Efrosinin D. Optimal control of queueing systems with heterogeneous servers. Queueing Systems, 2004, 46(3-4): 389-407

[64] Rykov V V. Monotone control of queueing systems with heterogeneous servers. Queueing Systems, 2001, 37(4): 391-403

[65] Jun-Wei Cao, Ke-Qin Li, Stojmenovic I. Optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers. IEEE Transactions on Computers, 2013, 63(1): 45-58

[66] Luenberger D G. Optimization by Vector Space Methods. New York: John Wiley & Sons, 1997

[67] Bolch G, Greiner S, de Meer H, Trivedi K S. Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications. Second Edition. Hoboken, New Jersey: Wiley-Interscience, American Statistical Association, 2006

[68] Tijms H C. A First Course in Stochastic Models. England: John Wiley & Sons, 2003

[69] Ivo A, Jacques R. Queueing Theory. Eindhoven: Department of Mathematics and Computing Science, Eindhoven University of Technology, 2001

[70] Smith J M G. Optimal design and performance modeling of M/G/1/K queueing systems. Mathematical and Computer Modelling, 2004, 39(9-10): 1049-1081

[71] Xiwei Qiu, Yuan-Shun Dai, Yan-Ping Xiang, Liu-Dong Xing. Correlation modeling and resource optimization for cloud service with fault recovery. IEEE Transactions on Cloud Computing, 2017, 7(3): 693-704

[72] Yi-Ju Chiang, Yen-Chieh Ouyang, Ching-Hsien Hsu. Performance and cost-effectiveness analyses for cloud services based on rejected and impatient users. IEEE Transactions on Services Computing, 2014, 9(3): 446-455

[73] Jagannatha S, Shravan N S, Kavya S. Cost performance analysis: Usage of resources in cloud using Markov-chain model//Proceedings of the 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS). Coimbatore, India, 2017: 1-8

[74] Grozev N, Buyya R. Multi-cloud provisioning and load distribution for three-tier applications. ACM Transactions on Autonomous and Adaptive Systems, 2014, 9(3): 1-21

[75] Hosseinalipour S, Huaiyu Dai. Options-based sequential auctions for dynamic cloud resource allocation//Proceedings of the 2017 IEEE International Conference on Communications (ICC). Paris, France, 2017: 1-6

[76] Yan-Zhi Wang, Shuang Chen, Goudarzi H, Pedram M. Resource allocation and consolidation in a multi-core server cluster using a Markov decision process model//Proceedings of the International Symposium on Quality Electronic Design (ISQED). Santa Clara, USA, 2013: 635-642

[77] Boots N K, Tijms H. An M/M/c queue with impatient customers. Top, 1999, 7(2): 213-220

[78] Nguyen B M, Tran D, Nguyen Q. A strategy for server management to improve cloud service QoS//Proceedings of the 2015 IEEE/ACM 19th International Symposium on Distributed Simulation and Real Time Applications (DS-RT). Chengdu, China, 2015: 120-127

[79] Bharkad N N, Durge M H. The application of queue theory in cloud computing to reduce the waiting time. International Journal of Engineering Research and Applications, 2014, 4(10): 64-69

[80] Shehabi A, Smith S, Sartor D, et al. United states data center energy usage report. Berkeley, California: Berkeley National Laboratory, United States Data Center Energy Usage Report: LBNL-1005775, 2004

[81] Briefing U. Senate. International energy outlook 2013. Washington, DC: US Department of Energy, International Energy Outlook 2013: DOE/EIA-0484, 2013

[82] Xinying Zheng, Yu Cai. Markov model based power management in server clusters//Proceedings of the IEEE/ACM International Conference on Green Computing and Communications and International Conference on Cyber, Physical and Social Computing. Hangzhou, China, 2010: 96-102

[83] Shuang Chen, Yan-Zhi Wang, Pedram M. A semi-Markovian decision process based control method for offloading tasks from mobile devices to the cloud//Proceedings of the 2013 IEEE Global Communications Conference (GLOBECOM). Atlanta, USA, 2013: 2885-2890

[84] Entezari-Maleki R, Sousa L, Movaghar A. Performance and power modeling and evaluation of virtualized servers in IaaS clouds. Information Sciences, 2017, 394: 106-122

[85] Zhou Zhou, Abawajy J, Chowdhury M, et al. Minimizing SLA violation and power consumption in cloud data centers using adaptive energy-aware algorithms. Future Generation Computer Systems, 2018, 86: 836-850

[86] Sayadnavard M H, Haghighat A T, Rahmani A M. Correction to: A reliable energy-aware approach for dynamic virtual machine consolidation in cloud data centers. The Journal of Supercomputing, 2019, 75(4): 2148-2148

WANG Shuang, Ph. D., lecturer. Her main interests focus on task scheduling, performance analysis, cloud computing, and reinforcement learning.

LI Xiao-Ping, Ph. D., professor, Ph. D. supervisor. His research interests include cloud computing and service computing.

CHEN Long, Ph. D., lecturer. His main interests include cloud computing, service computing.

Background

In cloud computing, cloud providers provide resources to providers by services. It is important to analyze performance of cloud service systems for service providers and consumers. Performance analysis aims at accessing the quantitative behavior of a system. Performance is regarded as a kind of quality of service in cloud computing which includes availability, throughput, reliability and so on. Performance contains latency and throughput. Latency describes the response time of requests while the throughput shows the maximum requests that the system currently handles. The more throughput implies more cost and energy consumption which requires quicker servers. Therefore, the cost, profit and energy consumption are predicted for service providers. Performance analysis supports the prediction for the consumers and service providers. With dynamic scenarios in cloud computing, it is difficult to obtain the performance in the system. Performance analysis is a kind of scheduling optimization problems that mapping stochastic requests to dynamic servers. The stochastic requests (tasks, jobs) with different types, various constraints, heterogeneous servers make the scheduling problem much more complex. With the stochastic property of requests, it is difficult to obtain the arrival patterns of requests. The distributions of the arrival time interval are various. The variable arrival and execution patterns make model construction much more complicated. The performance analysis procedures are different with different arrival patterns. The resources are dynamic and heterogeneous. With the increase of scale and complexity in cloud centers, performance analysis in cloud computing is becoming more difficult. Therefore, these are big challenges to analyze the performance for scheduling stochastic service requests in cloud computing.

Performance in cloud computing for stochastic requests has been analyzed with different scenarios but with strong constraints. In this paper, the performance analysis problems for stochastic service requests in cloud computing are surveyed. According to different arrival patterns, service patterns, queues and objectives, the performance analysis problems for stochastic requests in cloud computing are classified. The general framework for different queueing models is constructed. The procedures for performance analysis problems for stochastic service requests in cloud computing are proposed. Different models are compared and the promising topics with weak constraints are discussed. In future, we will focus on two topics: (1) performance analysis for stochastic requests in multiple queues in cloud computing, and (2) performance analysis with general patterns for requests or servers. This work is supported by the National Key Research and Development Program of China (No. 2017YFB1400800), the National Natural Science Foundation of China (No. 61872077), the Key Project of National Natural Science Foundation of China (No. 61832004) and the Collaborative Innovation Center of Wireless Communications Technology.