第 3 章 随机作业优化调度策略

CHAPTER 3

合理有效的作业调度是云计算系统性能优化和资源管理的前提。本章结合队列理论设计了一种细粒度的云计算系统模型。该模型由相互作用并协同工作的多个子模型构成，能够准确分析动态变化的云计算环境性能。本章提出了一种基于强化学习的用户作业调度算法，算法在各虚拟机资源占有量不变的约束下，以作业响应时间最小化为优化目标，通过作业的合理调度，提高了资源利用率和服务质量。实验结果不仅证明了该作业调度算法的有效性，而且深刻揭示了云计算环境中各性能指标，如达到率、服务率、虚拟机数量、队列长度等之间的关系。

3.1 引言

云计算是分布式计算、并行计算、效用计算、网络存储、虚拟化、负载均衡、热备份冗余等传统计算机和网络技术发展融合的产物。区别于传统的网络服务平台，云计算提供一种按使用量付费的服务模式，这种模式提供可用的、便捷的、按需的网络访问，进入可配置的计算资源共享池（资源包括网络、服务器、存储、应用软件、服务），这些资源能够被快速提供，只需投入很少的管理工作，或与服务供应商进行很少的交互。云计算的应用服务主要包括以下 3 种：基础设施即服务（Infrastructure-as-a-Service，IaaS）、平台即服务（Platform-as-a-Service，PaaS）和软件即服务（Software-as-a-Service，SaaS）。

如何在动态变化的云计算环境下合理有效地调度用户作业，提高资源利用率成为云计算服务必须解决的关键问题。虚拟化技术支持多个逻辑上独立的应用共享同一节点的物理资源，为提高节点利用率提供了可行的解决方案。但是，当前大多数数据

中心往往根据虚拟机对资源的峰值需求来决定集群的配置方案，而峰值需求远大于常态需求，所以仍难以有效提高资源利用率。

虚拟化技术将云计算物理资源等底层架构进行抽象，使得设备的差异和兼容性对上层应用透明，从而允许对云底层千差万别的硬件资源进行统一管理。此外，虚拟化技术简化了应用编写工作，使得开发人员可以仅关注业务逻辑，而无须考虑底层资源的供给与调度。通过虚拟化技术，单台物理服务器可以支持多个虚拟机运行多个操作系统和应用，这些应用服务驻留在各自的虚拟机上，形成一定的隔离，一个应用的崩溃不至于影响到其他应用的运行。最后，虚拟机的易于创建使得应用可以拥有更多的虚拟机来进行容错和灾难恢复，从而提高了自身的可靠性和应用性。但是，虚拟化技术的引入并没有减少云计算相关配置管理的复杂性。事实上，多虚拟机运行在同一物理计算基础设施上，增加了管理的难度。最终虚拟化技术仍然不能担保系统资源利用率，同时绝大部分现有工作仅能应用在小规模环境中而且难以扩展。

为了评估各种云计算平台性能，目前最常见的方式是采用队列理论或其他统计模型，其中响应时间是最重要也是最常用的服务等级协定（Service-Level-Agreement，SLA）性能指标之一。另外，市面上主要的云服务提供商（例如，亚马逊的 EC2、微软的 Azure 等）通常以小时为单位进行计费。因此无论不同的云计算模型侧重点有何不同，都需要在一定的资源占用率的约束下优化响应时间。然而目前文献中采用的主要方法是将整个云计算平台建模为单个队列模型，这种分析模型过于简单、难以扩展而且求解困难。

因此，深入研究云环境下作业调度机制不仅具有理论需要，而且具有重要的应用价值。本章的主要贡献总结为：首先设计了复杂云数据中心中不同运行阶段对应的不同子模型；其次根据队列理论分析了用户作业的响应时间；然后使用强化学习实现了用户作业优化调度算法；最后根据用户工作量和云平台编号，自适应调整系统资源。

本章剩余内容组织如下：3.2 节对相似研究进行介绍；3.3 节介绍了系统模型，包括作业调度子模型（Task Schedule Submodel，TSSM）、作业执行子模型（Task Execute Submodel，TESM）和作业传输子模型（Task Transmission Submodel，TTSM）；基于设计的系统模型，3.4 节中使用强化学习实现了作业调度算法，并使用状态聚合技术加速了算法收敛；3.5 节进行算法分析与比较；3.6 节对整章进行总结并指出接下来的研究方向。

3.2　国内外研究现状

作为目前最热门的研究领域，云计算吸引了大量科研人员的广泛关注，但使用严格的数学分析方法进行系统性能分析的还较少。当进行各种云计算平台性能分析时，目前最常见的方式是采用队列理论或其他统计模型，以响应时间作为最重要的性能衡量指标，而且响应时间也是最常用的服务等级协定性能指标之一。根据侧重点不同，使用队列理论进行系统性能分析的文献可被分为以下几类。

3.2.1　理论分析

使用队列理论进行云计算平台建模，最简单的模型为 $M/M/1$ 队列系统。但由于云环境下用户与服务提供商直接的动态交互，以及虚拟机资源的动态变化，$M/M/1$ 队列系统模型忽略了太多系统细节而无法直接使用。但该模型为队列理论为云计算系统分析奠定了理论基础。Yany 等假设用户作业到达间隔和服务时间均为指数分布，将云计算平台建模为 $M/M/m/m+r$ 队列系统，基于该系统模型分析得出了平均服务率、响应时间分布等重要指标的解析表达式。但该模型假设用户作业的等待时间、服务时间和执行时间等均为独立同分布的，这与实际情况不符。Liu 等采用马尔可夫请求队列模型，分析了虚拟机资源竞争和服务器宕机情况下云计算系统性能指标，而且该文献中没考虑到缓存队列长度的限制。

真实环境下的统计分析结果表明，云作业的到达间隔和服务时间通常不满足指数分布。Khazaei 等深入研究了这一问题。研究假设服务时间满足独立同分布，并将云平台建模为 $M/G/m/m+r$ 队列模型，基于该模型分析了云环境下的系统性能指标。

3.2.2　能耗管理

Yao 等针对大规模、地理分散的数据中心的路由决策和服务管理问题，将数据中心建模为 m 个平行和交互的队列系统，并设计了两阶段的前向规模控制算法进行费用和能耗优化。

3.2.3 资源分配

Gao 等针对云数据中心能耗和性能混合优化问题，设计了一种动态资源分配策略，通过动态的电压、频率调整和服务担保，降低了能耗并提高了应用层性能。Suresh 等设计了一种基于 $M/G/1$ 队列理论的资源分配算法，该算法通过云资源的回收机制，能够根据作业到达率的变化确保用户作业响应时间满足要求。相关工作也被扩展到多媒体云和大规模 Web 服务中。Tesauro 等首次将强化学习引入云计算资源分配之中，该算法的创新主要包括：

(1) 算法初始学习阶段使用预设值策略；

(2) 神经网络对 Q 值表的近似。

Julien 等针对用户作业、目标函数和基础设计，开发了一款通用的强化学习架构。Alexander 等针对大学数据中心，设计了一种弹性 Q 学习策略进行科学工作流调度。Bu 等设计了一种基于强化学习进行多层 Web 服务系统参数自配置算法。实验结果表明，该算法能够根据工作量和虚拟机资源自适应配置 Web 系统。

区别于目前的研究成果，本章设计了一种云计算环境下结合强化学习和队列理论的用户作业调度算法。算法引入相互交互的子模型，该模型涵盖了云数据中心的主要特性，包括用户请求达到率、虚拟化资源和弹性资源供给。通过对该模型的分析，可得到云计算系统的重要性能指标，例如，用户作业阻塞率、平均等待时间等。

3.3 系统模型

由于云计算平台的动态变化和复杂性，仅使用单队列模型很难进行系统分析和扩展。本章借鉴文献，重新设计了云计算系统模型，该模型由相互作用并协同工作的多个子模块构成，包括作业调度模块、作业执行模块和作业传输模块，核心部件作业分配器负责将用户作业分配到资源池中相应的计算服务器上进行执行，如图 3-1 所示。

3.3.1 作业调度子模块

在云计算环境中，用户通过网络提交作业服务请求并且接收作业执行结果。作业调度子模块(Task Schedule SubModel，TSSM)由一个有限大小的用户作业缓存队列

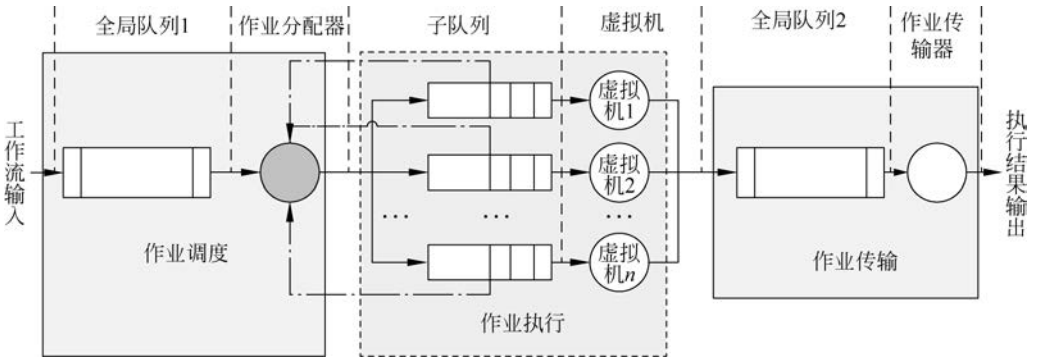

图 3-1　基于队列理论的细粒度云计算系统模型

和用户作业调度器构成。作业队列接收用户提交的作业请求并且维护一个先进先出全局作业队列。用户作业调度器将用户作业调度到作业执行子模块(Task Execute SubModel,TESM)相应的计算服务器上进行执行。虽然云计算系统的用户很庞大,但不同用户在相同时刻提交作业请求的概率很小,因此用户作业到达率可被描述为平均到达率为 λ^{tssm} 的泊松过程。

假设调度服务器的平均服务率为 μ^{tssm},则作业调度子模块可被建模为 $M/M/1$ 队列系统,当满足 $\lambda^{\text{tssm}}<\mu^{\text{tssm}}$ 条件时,用户作业的平均响应时间可表示为 $rt^{\text{tssm}}=\dfrac{1/\mu^{\text{tssm}}}{1-\dfrac{\lambda^{\text{tssm}}}{\mu^{\text{tssm}}}}$。

本章中采用了基于强化学习的任务调度器实现用户作业的优化调度,具体实现将在 3.4 节详细介绍。

3.3.2　作业执行子模块

作业执行子模块 TESM 从逻辑结构上可看成多台计算服务器相互并联而成。每台计算服务器均由相应的子任务缓存队列和作业执行器(虚拟机)构成。用户作业调度器将用户作业请求分配给某一具体的计算服务器的子任务缓存队列,作业执行器依次从子任务缓存队列中取出用户作业请求,执行完毕后将执行结果输送到作业传输子模块(Task Transmission SubModel,TTSM)。

本节假设所有的虚拟机同构并且具有相同大小的缓存空间 m,则作业执行子模块可被建模为一系列平行连接结构的 $M/M/1/m$ 队列系统。每台虚拟机以概率 p_i 接收

作业调度器分配的用户作业。当条件 $\lambda_i^{\text{tesm}}=p_i\lambda^{\text{tssm}}$ 和 $\sum_{i=1}^{N}p_i=1$ 满足时，队列系统稳定。假设虚拟机平均服务率为 μ^{tssm}，则第 i 个计算队列的响应时间可被表示为 $rt_i^{\text{tesm}}=\dfrac{1/\mu^{\text{tesm}}}{1-\dfrac{\lambda^{\text{tesm}}}{\mu^{\text{tesm}}}}$。考虑到用户作业可能被分配到任何一个作业队列，则用户作业平均响应时间可描述为 $rt^{\text{tesm}}=\sum_{i=1}^{n}p_i rt_i^{\text{tesm}}\sum_{i=1}^{n}p_i\dfrac{1/\mu^{\text{tesm}}}{1-\dfrac{\lambda_i^{\text{tesm}}}{\mu^{\text{tesm}}}}$。

3.3.3 作业传输子模块

作业传输子模块由一个全局作业执行结果队列和作业传输器组成。结果队列收到虚拟机发送的用户作业执行结果并维护一个先进先出全局结果队列。作业传输器从结果队列中取出作业执行结果并返回给用户。假设执行结果的到达率为 λ^{ttsm}，作业传输器的服务率为 μ^{ttsm} 并且满足条件 $\lambda^{\text{tssm}}<\mu^{\text{tssm}}$，与作业调度模块类似，作业传输模块同样可被建模为 $M/M/1$ 队列系统，平均响应时间可描述为 $rt^{\text{ttsm}}=\dfrac{1/\mu^{\text{ttsm}}}{1-\dfrac{\lambda^{\text{ttsm}}}{\mu^{\text{ttsm}}}}$。

综上所述，云计算环境下用户作业的平均响应时间可表示为 $rt^{\text{tot}}=rt^{\text{tssm}}+rt^{\text{tesm}}+rt^{\text{ttsm}}$。

3.4 基于强化学习的作业调度算法

作业调度器是用户作业执行序列和资源分配的控制器。一个设计合理的作业调度器不仅能够降低系统响应时间，而且能够提高资源利用率和系统吞吐量。因此云计算环境下的作业调度策略设计是云计算研究领域的一个重要课题。以著名的 Google 云计算平台为例，其 Hadoop 中的作业调度算法的研究方向主要包括数据本地化、任务推测机制以及异构环境等。

3.4.1 强化学习

强化学习属于机器学习的一个重要分支，它是智能体从环境状态到动作映射的学

习，以使动作从环境中获得的累计回报值最大。通常假定环境是马尔可夫型的，强化学习过程可以使用一个马尔可夫决策过程表示，其中状态转移概率和回报函数分别为 $P_a(s,s')=P_r(s_{t+1}=s' \mid s_t=s,a_t=a)$ 和 $R_a(s,s')=E(r_{t+1} \mid s_t=s,a_t=a,s_{t+1}=s')$。

智能体从当前状态 $s_t \in S$ 采取任一可选动作 $a_t \in A(s_t)$，然后转移到状态 s_{t+1} 并得到立即回报 r_t。动作选择不仅取决于立即回报，而且取决于将来可能得到的累计回报。本节采用 Q 学习算法实现用户作业调度。Q 值函数定义如下：

$$Q(s,a)=E\left\{\sum^{\infty}\gamma^k r_{t+k+1} \mid s_t=s,a_t=a\right\} \tag{3.1}$$

式中，$0<\gamma<1$ 为折扣因子。最优值函数 $Q^*(s,a)$ 定义如下：

$$Q^*(s,a)=\sum P_a(s_t,s_{t+1})(R_a(s_t,s_{t+1})+\gamma \max_{a_{t+1}} Q^*(s_{t+1},a_{i+1})) \tag{3.2}$$

式中，s_{i+1} 和 a_{t+1} 分别表示下一个状态和动作。

强化学习通过智能体的试错来发现最优行为策略 $\pi: S \to A$。优化策略 π^* 定义为任意初始状态下的最大累计回报，即

$$\pi^*(s)=\operatorname{argmax}(R_a(s_t,s_{t+1})+\gamma \sum P_a(s,s_{t+1})Q^*(s_{t+1},a_{t+1})) \tag{3.3}$$

3.4.2　基于强化学习的用户作业调度算法

正如 3.3 节所述，作业调度器实现用户作业从全局队列到任务执行队列的调度。假设云计算平台中某一时刻运行着的计算服务器的个数为 N，且每个服务器拥有的计算资源和计算能力相同，每台计算服务器的缓存队列容量均为 M，记计算服务器 i 的缓冲队列剩余容量为 s_i，则整个资源池中作业调度器可支配的剩余容量为 $0 \leqslant s_i \leqslant (M \times N)$。

作业调度器的工作过程可描述如下：假设 T_i 为第 i 个用户作业到达作业调度器时刻，作业调度器首先根据资源池中每个计算服务器缓冲队列剩余容量为 s_i、计算服务器当前作业执行时间以及等待作业的预计执行时间等参数做出作业调度决策，然后将第 i 个用户作业放置到某个计算服务器的缓存队列之中，最后作业调度器更新资源池中每个计算服务器缓冲队列剩余容量为 s_i-1 并等待下一个用户作业的到达。

由于云计算系统在整个运行过程之中，计算服务器会不停地执行用户作业，直到缓冲队列为空，所以资源池中每个计算服务器缓冲队列剩余容量是实时变化的，作业调度器需要定时更新资源池中每个计算服务器缓冲队列剩余容量。因此，作业调度器

的优化目标就是在资源池中同时运行多台计算服务器的环境下，合理调度用户作业，使其协同工作并实现在系统运行时段内用户作业完成率最大。该优化目标可表示为：

$$\underset{\left\{0\leqslant\sum_{i=0}^{N}s_i\leqslant M\times N\right\}}{\text{Minimize}}\quad rt^{\text{tot}}$$

subject to

$$0<\lambda^{\text{tssm}}<u^{\text{tssm}}$$

$$0<\frac{\lambda^{\text{tssm}}}{u^{\text{tssm}}}<1,\quad \forall i=1,2,\cdots,N$$

$$\sum_{i=1}^{n}p_i\lambda_i^{\text{tesm}}<u^{\text{tesm}},\quad \forall i=1,2,\cdots,N$$

$$0<\frac{\sum_{i=1}^{n}p_i\lambda_i^{\text{tesm}}}{u^{\text{tesm}}}<1,\quad \forall i=1,2,\cdots,N$$

$$0<\lambda^{\text{ttsm}}<u^{\text{ttsm}}$$

$$0<\frac{\lambda^{\text{ttsm}}}{u^{\text{ttsm}}}<1$$

$$(rt^{\text{tssm}}+rt^{\text{tesm}}+rt^{\text{ttsm}})\leqslant \text{SLA}$$

$$\sum_{i=1}^{n}p_i\lambda_i^{\text{tesm}}\leqslant\lambda^{\text{tssm}},\quad \forall i=1,2,\cdots,N \tag{3.4}$$

该作业调度算法满足马尔可夫决策过程，相应的概念可定义如下：

(1) 状态空间。对于云作业调度问题，状态可被定义为虚拟机的缓冲队列剩余容量 s_i，则对于整个云计算平台，状态空间可表示为 $s_i=(s_1,s_2,\cdots,s_n)$。

(2) 动作空间。对于作业调度器中的第 j 个用户作业请求，定义动作 $(0/1)_i^j$ 表示第 j 个用户作业分配到第 i 台虚拟机。因此云作业调度算法的动作空间可描述为向量形式，例如，$\boldsymbol{a}_i=(0,1,0\cdots,0)_i^2$，该向量表示用户作业请求分配到第 2 台虚拟机。

(3) 立即回报。本章以系统当前运行状态和作业调度效率作为立即回报。本章设计的回报函数基于以下两点考虑。

① 如果当前虚拟机的缓存队列大小为 M，则该用户作业将会被立即执行，等待时间和响应时间都为最小；

② 尽管当前虚拟机拥有很多空余缓存空间，若正在被执行的作业需要较长时间才能完成，则该队列中用户作业的等待时间和响应时间也会较长。本节设计的回报函数

如下：

$$r=\begin{cases}1, & \text{wt}<\overline{\text{wt}} \text{ 且 } s_i=\max(s_i)\\ 0, & \text{wt}<\overline{\text{wt}}\\ -1, & \text{其他}\end{cases} \tag{3.5}$$

式(3.5)中$\overline{\text{wt}}$表示平均等待时间。

Q学习通过与环境不断地交互和试错，利用环境反馈的评价信号实现决策的优化，其在线学习、免建模、以最大长期累计回报代替立即性能回报等特点使其成为强化学习的一个重要方法，特别适合于学习者对环境了解甚少和动态、复杂环境下策略的自适应学习，Q值表迭代更新过程如下：

$$Q(s_t,a_t)=Q(s_t,a_t)+a^*[r_{t+1}+\gamma^* Q(s_{t+1},a_{t+1})-Q(s_t,a_t)] \tag{3.6}$$

式(3.6)中α和γ分别表示学习率和折扣率。Q学习伪代码如算法3.1所示。

算法3.1　Q学习伪代码

```
1.  Initialize Q table
2.  Initialize state s_t
3.  error = 0
4.  repeat
5.    for each state s do
6.      a_t = get_action(s_t) using ε-greedy policy
7.        for (step=1; step<LIMIT; step++) do
8.          Take action a_t observe r and S_{t+1}
9.          Q_t = Q_t + a*(r+γ*Q_{t+1}-Q_t)
10.         error=MAX(error|Q_t-Q_{previous-t})
11.         s_t = s_{t+1}, a_{t+1}=get_action(s_t), a_t = a_{t+1}
12.       end for
13.   end for
14. until error<θ
```

3.4.3　状态简约

在多计算服务器系统的用户作业协同调度问题中，系统状态空间的大小会随着计算服务器个数的增加和缓存库容量的增加而呈指数形式增长，从而导致维数灾难，影响学习算法的收敛速度和优化效果。

假设每台计算服务器的缓存队列容量均为 M，记计算服务器 i 的缓冲队列剩余容量为 s_i，则有 $0 \leqslant s_i \leqslant M$。由于缓冲队列通常数目较大且实时变化，因此很难掌握其准确的数值，而且也没有必要掌握其精确数值。定义抽象函数 $\phi: S_i \rightarrow E_i$，则对任意的 s_i，$\phi(s_i) \in E_i$ 为聚类后的一个抽象状态。抽象反函数 $\{\phi^{-1}(e_i) \mid e_i \in E_i\}$ 将原始状态空间 S_i 划分成多个子类，即聚类后的抽象状态空间。本节根据缓存队列剩余容量的大小，将原始状态空间划分成 5 个抽象状态，映射过程如下：

$$E=\begin{cases} S_{\mathrm{v}}, & s_i=0 \\ S_{\mathrm{l}}, & s_i \in \left(1, \dfrac{M}{3}\right) \\ S_{\mathrm{m}}, & s_i \in \left(\dfrac{M}{3}+1, \dfrac{2M}{3}\right) \\ S_{\mathrm{h}}, & s_i \in = \left(\dfrac{2M}{3}+1, M-1\right) \\ S_{\mathrm{f}}, & s_i=M \end{cases} \tag{3.7}$$

式(3.7)中下标 v、l、m、h、f 分别表示剩余缓存队列的状态为空(缓存队列已满)、有较少缓存空间、中等缓存空间、较多缓存空间、全空。因此，经过状态聚类后，抽象状态空间为 $E_i=\{S_{\mathrm{v}}, S_{\mathrm{l}}, S_{\mathrm{m}}, S_{\mathrm{h}}, S_{\mathrm{f}}\}$。设对于第 i 个用户作业请求，作业调度器采取的动作只与聚类状态 S_i 有关，记为 $a_i(S_i)$。显然系统存在如下两种状态，其动作具有特殊性。

(1) 队列空 S_{v}。作业调度器拒绝将用户作业请求分配给剩余缓存队列为空的作业执行器，即 $a_i(S_{\mathrm{v}})=0$。

(2) 队列满 S_{f}。作业调度器优先将用户作业请求分配给剩余缓存队列为满的作业执行器，即 $a_i(S_{\mathrm{f}})=1$。

对于其他的状态空间 $E_i=\{S_{\mathrm{l}}, S_{\mathrm{m}}, S_{\mathrm{h}}\}$，作业调度的优先级依次增大。

对于第 i 个用户作业请求，作业调度器的策略可记为 $v_i(a(s_{\mathrm{v}}), a(s_{\mathrm{l}}), a(s_{\mathrm{m}}), a(s_{\mathrm{h}}), a(s_{\mathrm{f}}))$，状态转移记为 $(e_i(t_i), a_i(e_i(t_i)), e_i(t_{i+1}))$，对应一个观测状态样本 $<s_i(T_n), a_i(e_i(t_i)), s_i(T_{n+1}), w_i(T_n), \mu_i(T_n)>$，状态转移的即时回报 $r(e_i(t_i), a_i(e_i(t_i)), e_i(t_{i+1}))$，则状态聚类下的 Q 学习伪代码如算法 3.2 所示。

算法 3.2　状态聚类下的 Q 学习伪代码

```
1.  Initialize State aggregation φ: S_i → E_i
2.  Initialize Q table
3.  error = 0
4.  repeat
5.    for each abstract state E_i do
6.      Update Q-value table using Algorithm 1
7.    end for
8.  until error<θ
```

3.5　性能评估

根据采用的云计算平台模型和设计的作业调度算法，分别在 Matlab 和 CloudSim 平台上进行性能验证。课题组开发了基于离散事件驱动的数值仿真器，实现了本章设计的基于强化学习的用户作业调度算法 Q-sch，并与流行的静态作业调度算法（平均作业调度算法 Equ-sch）、动态作业调度算法（随机作业调度算法 Ran-sch、混合作业调度算法 Mix-sch）等进行比较。

3.5.1　仿真云平台实验验证

1. 不同作业到达率下的响应时间比较

图 3-2 所示为用户作业到达率从 10 个/秒增加到 20 个/秒，各种作业调度算法响应时间的比较结果。图 3-3 所示为作业服务率从 1 个/秒增加到 5 个/秒，各种作业调度算法响应时间的比较结果，实验结果均证明本课题设计算法优于对比算法。

2. 不同服务率下的响应时间比较

图 3-4 为相同用户作业到达率和虚拟机服务率下，各种算法虚拟机资源利用率和负载均衡的差异，从实验结果可见，本章算法不仅能够提高虚拟机资源利用率，而且实现了各虚拟机的负载均衡。

3. 不同缓存空间大小下的响应时间比较

图 3-5 为不同缓存大小（队列长度）对用作业响应时间的影响，实验结果表明各种

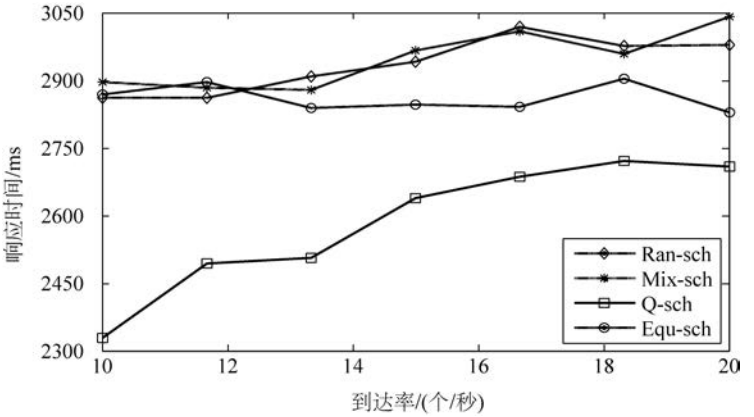

图 3-2　不同到达率下作业响应时间比较

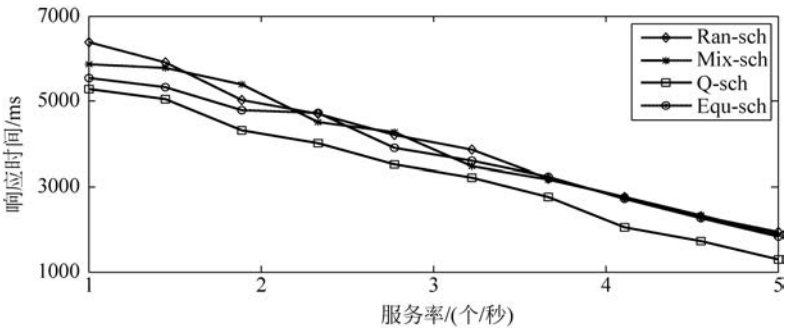

图 3-3　不同服务率下作业响应时间比较

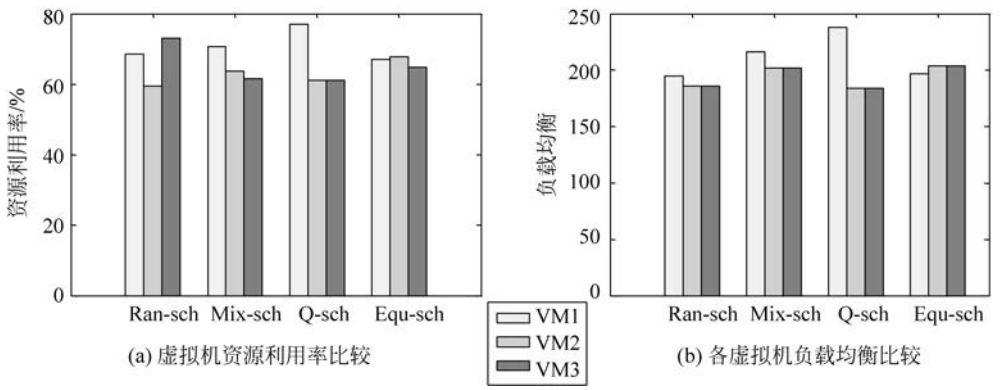

图 3-4　相同到达率和服务率下各虚拟机性能分析

作业调度算法均对缓存大小(队列长度)不敏感。

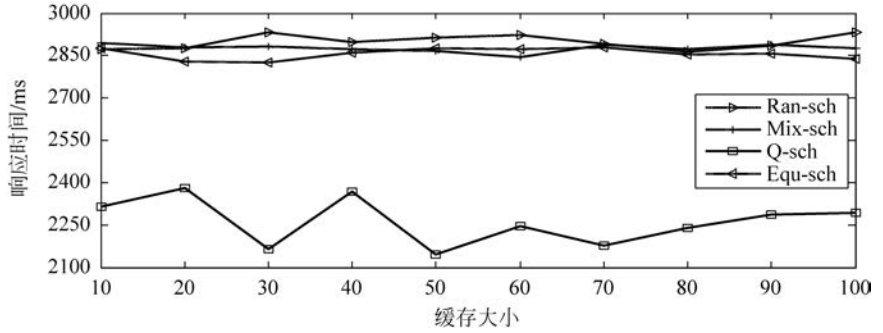

图 3-5　不同缓存下用户作业响应时间比较

4. 不同作业到达率和服务率下的响应时间比较

图 3-6 为用户作业响应时间随作业到达率和虚拟机服务率变化情况,实验结果表明,用户作业响应时间随作业到达率增加和服务率降低呈指数增长趋势。从该实验结果表明,可从以下两个方面优化云平台性能:根据用户作业到达率动态调整系统资源,实现虚拟机服务率的自适应调整以避免违反服务等级协议;在确定的系统资源约束下,可以动态调整用户作业到达率,以实现降低作业响应时间,提高云平台服务质量。

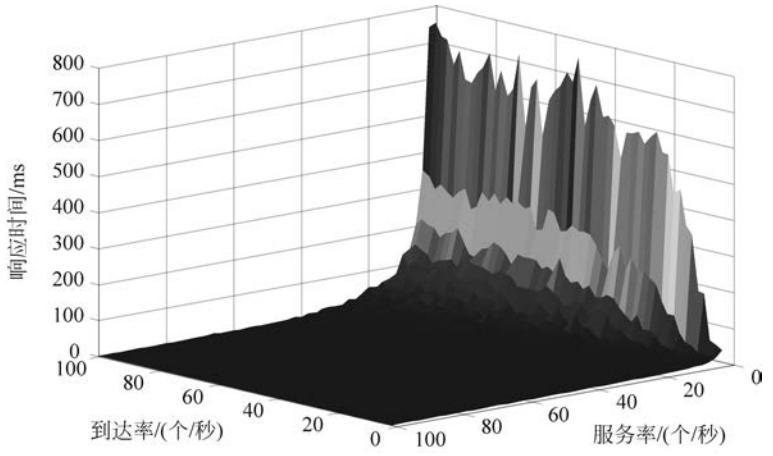

图 3-6　响应时间随到达率和服务率变化情况

3.5.2 真实云平台实验验证

根据用户作业到达情形和响应要求的不同，在标准云计算测试平台 CloudSim 上分别进行在线作业调度和离线作业调度验证。在线作业调度为用户作业连续到达，要求作业调度算法具有实时响应能力和低的时间复杂度；离线作业调度为用户作业批到达，要求作业调度算法能够根据用户作业对系统资源的需求和系统运行状态进行优化调度。本章设计的基于强化学习的用户作业调度算法 Q-sch，并与流行的先进先出作业调度算法 FIFO-sch、公平作业调度算法 Fair-sch、贪婪作业调度算法 Greedy-sch、随机作业调度算法 Ran-sch 等进行比较，选用平均响应时间作为衡量指标。

1. 在线作业调度算法

从图 3-7 的实验结果可得到以下结论，首先在相同的云资源供给条件下，各种作业调度算法的平均响应时间均随着作业数量的增多而增大；其次各种作业调度算法的响应时间均随着作业数量的增多而逐渐趋向汇聚；最后证明课题组算法优于对比算法。在线作业调度算法参数设置如表 3-1 所示。

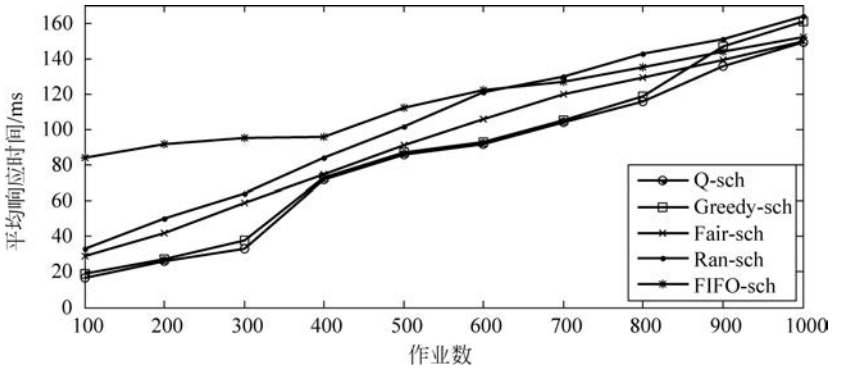

图 3-7 不同作业数下的平均响应时间比较

表 3-1 在线作业调度算法参数设置

参　　数	取　　值
作业长度	$(1\sim2)\times10^{10}$ 条指令
作业总数	100～1000
虚拟机总数	5
虚拟机频率	每秒$(1\sim3)\times10^{9}$ 条指令

续表

参　　数	取　　值
虚拟机内存	512～2048MB
虚拟机带宽	500～1000MB/s
虚拟机缓存	10～50
物理机数量	5
数据中心数量	1
主机数量	1

2. 离线作业调度算法

图 3-8 为本章设计的基于强化学习的用户作业调度算法 Q-sch，并与流行的遗传作业调度算法 SGA-sch、改进的遗传作业调度算法 MGA-sch 等进行比较，选用平均完成时间作为衡量指标。图 3-8 的实验结果同样证明课题组算法的优越性。离线作业调度算法参数设置如表 3-2 所示。

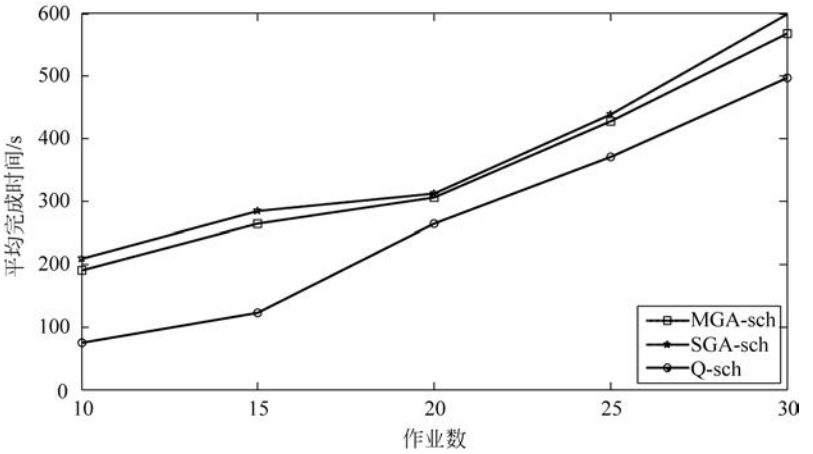

图 3-8　不同作业数下的平均完成时间比较

表 3-2　离线作业调度算法参数设置

参　　数	取　　值
作业长度	$(1\sim10)\times10^{10}$ 条指令
作业数量	10～30
虚拟机数量	10
虚拟机频率	每秒$(1\sim5)\times10^{8}$ 条指令

续表

参　数	取　值
虚拟机内存	512～2048MB
虚拟机带宽	500～1000MB/s
虚拟机缓存	10～50
物理机数量	1 或 2
数据中心数量	2
主机数量	2

3.6 小结

本章深入研究了云计算平台中用户作业的优化调度问题，设计了基于队列理论的云计算平台模型，该模型由相互连接的 3 个子模块组成并且能够刻画云计算服务过程。通过该模型，分析了每个子模块的响应时间，设计了一种新颖的基于强化学习的用户作业调度算法。仿真和真实云计算环境下的实验结果表明，本章算法不仅能够提高资源利用率，而且揭示了系统性能指标直接的隐含关系。

今后打算从以下几个方面进行扩展研究：协同作业调度和资源供给，期望进一步降低响应时间；深入研究云实例，考虑虚拟机宕机、迁移、通信费用等多种因素对调度算法的影响等。