Archive of SID

زمان‌بندی آینده‌نگرانه وظایف در محاسبات ابری مبتنی بر یادگیری تقویتی

سید اکبر مصطفوی[۱]، فاطمه احمدی[۲]، مهدی آقا صرام[۳]
۱- گروه مهندسی کامپیوتر – دانشگاه یزد – یزد – ایران – a.mostafavi@yazd.ac.ir
۲- گروه مهندسی کامپیوتر – دانشگاه یزد – یزد – ایران –fa.ahmadi@stu.yazd.ac.ir
۳- گروه مهندسی کامپیوتر – دانشگاه یزد – یزد – ایران – mehdi.sarram@yazd.ac.ir

چکیده: با پیدایش رایانش ابری، کاربران منابع محاسباتی را بر اساس قانون پرداخت در ازای استفاده از ارائه‌دهندگان سرویس ابری دریافت می کنند. با توجه به محدودیت و پویایی منابع ابری برای اجرای درخواست‌های متنوع و متغیر با زمان کاربران، نیازمند یک مکانیزم زمان‌بندی مؤثر هستیم که خود را با شرایط پویای سیستم وفق داده و بهره‌وری منابع و رضایت کاربران را در بلندمدت تأمین نماید. الگوریتم‌های زمان‌بندی موجود تلاش می‌کنند با توجه به وضعیت فعلی سیستم، زمان‌بندی بهینه را بیابند که با توجه به پویایی درخواست‌های کاربران و عدم درک صحیح زمان‌بند از محیط لزوماً منجر به دستیابی به نتیجه بهینه در بلند مدت نمی‌گردند و غالباً این الگوریتم‌ها توانسته‌اند تنها یکی از پارامترهای کیفیت سرویس را بهبود ببخشند. در این مقاله یک روش زمان‌بندی مبتنی بر روش یادگیری تقویتی پیشنهاد می‌شود که به دلیل قابلیت تطبیق با محیط و ارائه پاسخ مناسب به درخواست‌های متغیر با زمان، با تخصیص آینده‌نگرانه وظایف به منابع منجر به افزایش کارایی سیستم در بلندمدت می‌گردد. نتایج این مقاله نشان می‌دهد که روش پیشنهادی نه تنها منجر به کاهش زمان پاسخ، زمان انتظار و زمان تکمیل کار می‌گردد بلکه نرخ بهره‌وری منابع را هم به عنوان هدف فرعی افزایش می‌دهد. روش پیشنهادی در محدوده تعداد وظایف بالا، زمان پاسخ را به طور میانگین حدود ۴۹/۵۲ درصد نسبت به Random، ۴۶/۰۳ نسبت به Mix، ۴۳/۹۹ نسبت به FIFO، ۴۳/۵۳ نسبت به Greedy و ۳۸/۶۸ درصد نسبت به Q-sch بهبود ببخشیده است.

واژه‌های کلیدی: محاسبات ابری، زمان‌بندی وظایف، یادگیری تقویتی، زمان‌بندی آینده‌نگرانه وظایف، زمان پاسخ.

Reinforcement-Learning-based Foresighted Task Scheduling in Cloud Computing

Seyedakbar Mostafavi[1], Fatemeh Ahmadi[2], Mehdi Agha Sarram[3]

1-Department of Computer Engineering, Yazd University, Yazd, Iran, Email: a.mostafavi@yazd.ac.ir
2- Department of Computer Engineering, Yazd University, Yazd, Iran, Email: fa.ahmadi@stu.yazd.ac.ir
3- Department of Computer Engineering, Yazd University, Yazd, Iran, Email: sarram@yazd.ac.ir

Abstract: With the apperance of cloud computing, users receive computing resources according to pay as you go of cloud service provider. An optimized scheduling approach for mapping all the tasks to the resources is an essential problem due to the limitations and dynamics of resources for requests which vary during the time. This solution may lead to improvement of system's efficiency. There are different methods for cloud computing scheduling with different parameters such as response time, makespan, waiting time, energy consumption, cost, utilization rate, and load balancing. But many of these methods are not suitable for improving scheduling performance in a condition that users requests change during the time. So in this thesis a scheduling method based on reinforcement learning is proposed. Adopting with environment conditions and responding to unsteady requests, reinforcement learning can cause a long-term increase in system's performance. The results show that this proposed method can not only reduce the response time and makespan but also increase resource efficiency as a minor goal. Our proposed illustrates improvements in response time for 49.52%, 46.03%, 43.99%, 43.53% and 38.68% over Random, Mix, FIFO, Greedy and Q-sch algorithms, respectively.

Keywords: Cloud computing, Task scheduling, Reinforcement learning, Foresighted task scheduling, Response time.

تاریخ ارسال مقاله: ۱۳۹۷/۰۱/۲۶
تاریخ اصلاح مقاله: ۱۳۹۷/۰۵/۰۶
تاریخ پذیرش مقاله: ۱۳۹۷/۰۷/۲۳
نام نویسنده مسئول: سید اکبر مصطفوی
نشانی نویسنده مسئول: ایران – یزد – صفائیه – بلوار دانشگاه – دانشگاه یزد – پردیس فنی مهندسی – گروه مهندسی کامپیوتر

www.SID.ir

Archive of SID

۱- مقدمه

محیط ابری امکان دسترسی به منابع زیادی را بر اساس نیاز کاربران فراهم می‌کند و کاربران می‌توانند به صورت انعطاف‌پذیر و مقیاس‌پذیر از آن‌ها استفاده کنند. تعداد کاربران که از خدمات ابر استفاده می‌کنند روز به روز در حال افزایش هستند، بنابراین با توجه به پویایی درخواست کاربران، وجود الگوریتم زمان‌بندی کارا و مؤثر که مسئول تخصیص مناسب وظایف به منابع و افزایش عملکرد سیستم گردد، بسیار احساس می‌شود [۱]. زمان‌بندی اشاره به نگاشت وظایف به منابع خاص به منظور بهبود عملکرد سیستم و کیفیت خدمات دارد که پارامترهای متفاوتی مانند حداقل زمان اتمام کار، پشتیبانی از توافقات سطح سرویس، هزینه، امنیت، قابلیت اطمینان را در نظر می‌گیرد. دستیابی به کیفیت سرویس مطلوب یکی از موضوعات چالش برانگیز روش‌های زمان‌بندی می‌باشد. امروزه الگوریتم‌های اکتشافی و فرا اکتشافی متعددی در این زمینه پیشنهاد شده است که تا حدی توانسته‌اند برخی مشکلات را برطرف کنند ولی لزوماً منجر به بهبود عملکرد سیستم در بلند مدت نمی‌شوند.

مقاله [۲] یک مدل زمان‌بندی بهینه مبتنی بر روش یادگیری تقویتی با استفاده از تقسیم مسئله به چند ریزمدل و تئوری صف ارائه می‌دهد. در این روش ارسال کننده وظایف با به کارگیری روش یادگیری کیو تخصیص مناسبی از وظایف ایجاد می‌کند که منجر به بهبود زمان پاسخ در تخصیص وظایف به منابع می‌شود. همچنین برای غلبه بر مشکل افزایش فضای حالت از روش تجمع حالت استفاده می‌کند که سرعت همگرایی برنامه را بالا می‌برد. این مقاله با اینکه زمان پاسخ را کاهش می دهد ولی به دلیل نداشتن درک درستی از محیط منجر به عدم دستیابی به حداقل زمان پاسخ برای وظایف پویا گردیده است. مقاله [۳] به منظور پشتیبانی از مقیاس‌پذیری برنامه‌ها در ابر از روش یادگیری تقویتی به منظور بهینه‌سازی تخصیص منابع استفاده می‌کند.این زمان‌بندی با روش یادگیری کیو عامل را به یادگیری سیاست‌های بهینه برای تخصیص منایع در زیر ساخت ابر در حالیکه هیچ دانشی از محیط ندارد، سوق می دهد. الگوریتم زمان‌بندی را با وجود درخواست‌های متنوع کاربران و عدم اطمینان از عملکرد صحیح سیستم ارائه می‌دهد که با موازی کردن فرایند یادگیری، زمان را برای رسیدن به سیاست مطلوب کاهش می دهد. با این وجود یک رویکرد سطح بالاتری در رابطه با یادگیری برای حل مقیاس پذیری برای پشتیبانی از عملکرد سیستم مانند زمان پاسخ برنامه‌ها مورد نیاز است. مقاله [۴] یک سناریو زمان‌بندی وظایف در محیط ابر و راه حل‌های موجود در آن را معرفی می‌کند. راه حل مورد استفاده در این مقاله دسته‌بندی وظایف بر اساس یک ویژگی اقتصادی به نام بهره‌وری است که هر وظیفه را با ویژگی بهره‌وری توسط تابعی مقداردهی و توصیف می‌کند. این الگوریتم با در نظر گرفتن نرخ شکست و نرخ بازیابی منابع، فرایند زمان‌بندی وظایف در ابر را مدل سازی می کند و می‌تواند وضعیت سیستم در آینده را با بکارگیری روش یادگیری تقویتی در نظر بگیرد. نتایج تجربی نشان می‌دهد این الگوریتم زمان‌بندی مبتنی بر روش یادگیری تقویتی از لحاظ بهره‌وری بسیار سازنده و منجر به بهینه‌سازی بهره‌وری سیستم ابری با در نظر گرفتن پارامتر قابلیت اطمینان می‌شود. این مقاله تنها پارامتر بهره‌وری را مورد بررسی قرار داده است و به دیگر پارامترهای کیفیت سرویس مانند زمان پاسخ، زمان تکمیل کار توجهی نداشته است.

بهینه سازی بهره‌وری انرژی یکی دیگر از چالش‌های روش‌های زمان‌بندی است که مقاله [۵] به طور کارا یک چهارچوب مدیریت انرژی مؤثر مبتنی بر روش‌های یادگیری تقویتی با در نظر گرفتن ناهمگنی موجود در منابع و متفیر بودن درخواست کاربران مدل سازی می‌کند، همچنین با در نظر گرفتن نیازهای متفاوت وظایف به منابع محاسباتی و اولویت‌های آنها با استفاده از تکنیک گروه‌بندی وظایف، نگاشت مطلوبی از منابع با ظرفیت‌های متفاوت برای اجرای وظایف ایجاد می‌کند. این روش زمان‌بندی مدیریت انرژی را به صورت موثری کاهش می‌دهد و همچنین منجر به بهبود زمان پاسخ می‌گردد. امروزه تقاضاهای زیادی برای سیستم‌های توزیع شده مانند گرید و ابر وجود دارد. در حالیکه سیستم‌های توزیع شده قدرت محاسباتی بالایی را فراهم می‌کنند اما تضمین قابلیت اطمینان در آنها دشوار است که در مقاله [۶] یک روش زمان‌بندی جدید مبتنی بر یادگیری تقویتی با هدف ارائه سرویس قابل اعتماد ارائه می‌شود که تمرکز بر روی بهبود زمان اجرا با پیچیدگی محاسباتی پایین دارد. استفاده از رویکرد یادگیری تقویتی در مدیریت منابع موجب می‌شود، زمان‌بند که نقش عامل را در یادگیری ایفا می‌کند با در نظر گرفتن تنوع در منابع و وظایف، با توجه به حالات مختلف، خود را با تغییرات پویای محیط وفق دهد. این روش با آگاهی به دسترس پذیری منابع و دریافت بازخورد از محیط به طور قابل ملاحظه‌ای به راه حل بهینه نزدیک می‌گردد.

با وجود مطالعات وسیع در زمینه زمان‌بندی وظایف در محیط ابری، فقط تعداد محدودی از آنها بر روی عملکرد ترکیبی کارها با محاسبات سنگین و کارها با داده‌های فشرده تمرکز دارند که مقاله [۷] یک طرح زمان‌بندی ترکیبی جدید مبتنی بر یادگیری تقویتی با هدف ارائه زمان‌بندی مؤثر کارها تحت محدودیت منابع و توافق نامه سطح سرویس به منظور به حداقل رساندن زمان تکمیل کار ارائه می‌دهد، همچنین از تکنیک موازی سازی برای تعادل بین اکتشاف و استخراج در فرایند یادگیری و سرعت بخشیدن به سرعت همگرایی الگوریتم یادگیری کیو استفاده کرده است. نتایج حاصل از شبیه سازی نشان می‌دهد که طرح پیشنهادی می‌تواند از منابع ابری به طور بهینه و متعادل در جهت به حداقل رساندن زمان تکمیل کار تحت محدودیت منابع استفاده کند. این مقاله با وجود بررسی زمان تکمیل کار به بررسی دقیق و تحلیل زمان پاسخ و زمان انتظار نپرداخته است. ارائه دهندگان سرویس ابری، مراکز داده را در اختیار کاربران قرار می‌دهند و سود آنها بر اساس سطح دسترسی کاربران به سرویس‌ها و اجاره ماشین‌های مجازی برآورد می شود، همچنین کاهش هزینه‌های انرژی منجر به افزایش سود می‌گردد.

www.SID.ir

Archive of SID

یکی از چالش‌های اصلی برای ارائه دهندگان سرویس، به حداقل رساندن هزینه‌های انرژی در مراکز داده است. مقاله [۸] با در نظر گرفتن مقیاس پذیری و وابستگی وظایف، یک طرح زمان‌بندی جدید مبتنی بر یادگیری تقویتی عمیق ارائه می‌دهد که قادر به کاهش هزینه‌های انرژی در مقیاس‌های بالا با تعداد سرور و درخواست کاربران متعدد می‌باشد. این روش در دو مرحله تأمین منابع و زمان‌بندی وظایف به طور خودکار بهنرین تصمیمات را در بلند مدت با توجه به تغییر محیط در رابطه با درخواست کاربران می‌گیرد. این مقاله تنها به مسئله کاهش هزینه انرژی پرداخته است و دیگر پارامترهای کیفیت سرویس را در نظر نگرفته است. چگونگی زمانبندی منابع به صورت پویا و تضمین کیفیت خدمات در بلند مدت یک مسئله کلیدی در محیط ابری است که مقاله [۹] یک روش زمان‌بندی منابع پویا براساس سیستم تست خودکار با استفاده از روش یادگیری تقویتی برای به حداکثر رساندن عملکرد سیستم در بلند مدت در سیستم محاسبات ابری ارائه می‌دهد که منجر به افزایش در خواست کاربران، افزایش نرخ بهره برداری منابع و همچنین کاهش هزینه‌ها و افزایش سود می‌گردد. این مقاله با ایجاد تغییراتی در الگوریتم یادگیری تقویتی منجر به بهبود عملکرد آن نسبت به روش یادگیری تقویتی مبتنی بر روش شبه حریصانه شده است که در در بیشتر موارد عملکرد و کارایی بهتری در تأمین منابع داشته است. این الگوریتم باید در مقیاس‌های بزرگ‌تر با روش‌ها و وظایف بیشتری نیز تست شود، همچنین با توسعه آن بتوان یک الگوریتم سریع‌تر و یا نزدیک به زمان واقعی ارائه داد که بتواند دیگر پارامترها را نیز بررسی کند. مقاله [۱۰] یک الگوریتم چند هدفه و عادلانه مبتنی بر روش شاخه و حد برای تخصیص وظایف در ابر سیار ارائه می‌دهد. این الگوریتم مجموعه حواب های پارتو را به دست می‌آورد و از روشی مبتنی بر مفهوم فاصله از جواب‌های ایده‌آل مثبت و منفی برای انتخاب جواب نهایی استفاده می کند. روش‌های مکاشفه‌ای با وجود دستیابی به جواب بهینه، بهینگی قطعی را تصمین نمی‌کنند، در حالیکه روش ارائه شده در این مقاله مذکور با در نظر گرفتن زمان واگذاری و انرژی مصرفی سعی در توزیع مناسب وظایف به همراه کمینه کردن همزمان دو معیار زمان و انرژی دارد. این الگوریتم خرابی و مشکلات احتمالی در حین واگذاری و همچنین در دسترس بودن گره‌ها برای تخصیص وظایف را در نظر نگرفته است و جواب بهینه را برای بلند مدت تضمین نمی‌کند.

مقاله [۱۱] با استفاده از الگوریتم رقابت استعماری روشی جدید و کارا برای مکان یابی ماشین‌های مجازی بر روی ماشین‌های فیزیکی به منظور کاهش مصرف انرژی و کاهش اتلاف منابع پیشنهاد داده است. نتایج نشان می‌دهند روش ارائه شده در مقایسه با دیگر الگوریتم‌ها به جواب‌های قابل قبولی دست پیدا کرده است. این الگوریتم برخی محدودیت‌های منابع ابری و پویایی محیط ابری را در نظر نگرفته است ضمن اینکه با وجود کاهش مصرف انرژی و کاهش اتلاف منابع به بررسی دو معیار مهم زمان تکمیل کار و زمان پاسخ نپرداخته است.

در اکثر الگوریتم‌های زمان بندی معیارهای متفاوتی مانند زمان پاسخ، زمان تکمیل کار، انرژی، توازن بار، هزینه، قابلیت اطمینان در نظر گرفته شده است که تا حدی توانسته‌اند برخی از آنها را برطرف کنند ولی با توجه به دسترسی محدود به اطلاعات در مورد وظایف لزوماً منجر به بهبود زمان پاسخ و زمان تکمیل کار توام با افزایش نرخ بهره‌وری در بلندمدت نمی‌گردند. در این مقاله با استفاده از روش یادگیری تقویتی به دنبال ارائه راهکارهای آینده‌نگرانه‌ای هستیم که با توجه به پویایی محیط ناشی از درخواست‌های متنوع و متغیر با زمان، منافع سیستم محاسبات ابری را در بلندمدت بیشینه کند که هدف اصلی آن کاهش زمان پاسخ و زمان تکمیل کار به همراه افزایش نرخ بهره‌وری سیستم است. روش پیشنهادی با کاهش حداکثری زمان انتظار وظایف در صف بافر ماشین‌های مجازی توانسته است در محدوده تعداد وظایف بالا به طور میانگین ۵۱/۱۷ درصد زمان انتظار، ۴۴/۳۵ درصد زمان پاسخ و ۲۰/۳۰ درصد زمان تکمیل کار را نسبت به سایر الگوریتم‌های زمان‌بندی (Q-sch-Random-Mix-FIFO-Greedy) بهبود ببخشد.

۲- مدل سیستم

در این بخش یک طرح جدید زمان‌بندی وظایف بر اساس یادگیری تقویتی ارائه می‌شود، نقش عامل را زمان‌بند یا ارسال‌کننده وظایف ایفا می‌کند. با تجربه‌ای که زمان‌بند در تعامل با محیط از طریق یادگیری به دست می‌آورد، تصمیمات بهتری می‌گیرد و به اهدافی دست پیدا می‌کند که به طور مستقیم بهینه‌سازی آن‌ها مشکل است. زمان‌بند با دریافت پاداش‌ها و مجازات‌ها در طول برنامه و بیشینه‌سازی پاداش‌های جمع‌آوری شده مورد انتظار در طول زمان در تخصیص مناسب وظایف به ماشین‌های مجازی موفق می‌گردد و در نهایت منجر به کاهش زمان پاسخ و زمان اتمام کار به همراه افزایش نرخ بهره‌وری سیستم می‌گردد. با توجه به شکل ۱ مدل سیستم از سه بخش کلی ارسال وظایف، تخصیص وظایف و اجرای وظایف تشکیل شده است.

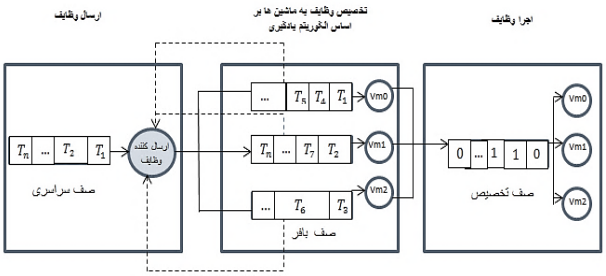
در بخش ارسال، زمان‌بند وظایف کاربران را از صف سراسری به یکی از زیرصف‌های ماشین مجازی (Virtual Machine) اختصاص می‌دهد. صف سراسری درخواست‌های متفاوت کاربران را می‌پذیرد و به جهت آنکه اولین درخواست ورودی، اولین درخواست خروجی باشد، آن‌ها را در صف قرار می‌دهد و به ترتیب برای پردازش انتخاب می‌کند. هر VM دارای یک صف بافر است که ظرفیت آن ثابت است. ارسال‌کننده وظایف در هنگام تخصیص وظایف به ماشین‌ها، از ظرفیت بافر هر VM اطلاع دارد. اگر فرض شود تعداد VM ها برابر K و ظرفیت بافر هر VM برابر N باشد، ظرفیت اشغال شده بافر هر VM با S_i مشخص می‌شود به طوریکه کل ظرفیت برای زمان‌بند برابراست با :

$$0 \le \sum_{i=0}^{k} S_i \le K \times N \qquad (۱)$$

www.SID.ir

Archive of SID

با توجه به شکل ۱ ابتدا زمان‌بند، به ترتیب وظایف را از صف سراسری انتخاب می‌کند. با انتخاب هر وظیفه، زمان‌بند با حالتی از محیط برخورد می‌کند که با قرار گرفتن در این حالت به صورت تصادفی یا شبه حریصانه با استفاده از روش یادگیری کیو (یکی از روش‌های یادگیری تقویتی است،) یکی از ماشین‌های مجازی موجود را برای نگاشت آن وظیفه انتخاب می‌کند. با نگاشت هر درخواست به VM به ازای انجام عمل منتخب (انتخاب یکی از VM ها) پاداش دریافت می‌کند و ارزش عملش را با توجه به حالتی که درآن قرار دارد و پاداشی که دریافت کرده است، محاسبه می‌کند. بعد از هر تخصیص وظیفه به ماشین مجازی، مقدار ظرفیت بافر ماشین‌ها بروز رسانی می‌شوند. با ورود درخواست بعدی، زمان‌بند حالت جدیدی از محیط را ملاقات می‌کند و مراحل قبل به صورت مکرر انجام می‌شود تا زمانی که زمان‌بند مطمئن شود کل وظایف به بهترین شکل به VM ها تخصیص داده شده‌اند، در نهایت با ایجاد نگاشت مناسبی از وظایف، آن‌ها را برای اجرا به صف بعدی (صف تخصیص) در فاز اجرا منتقل می‌کند، این صف آرایه‌ای است که اندیس‌های آن شماره وظایف و مقدار آن شماره VM ای است که قرار است وظیفه به آن تخصیص پیدا کند. این آرایه مشخص می‌کند که کدام وظیفه به کدام ماشین مجازی اختصاص یابد به طوریکه درخواست‌ها کمترین زمان انتظار را در صف بافر ماشین‌های مجازی متحمل گردند که منجر به کاهش زمان پاسخ شود.

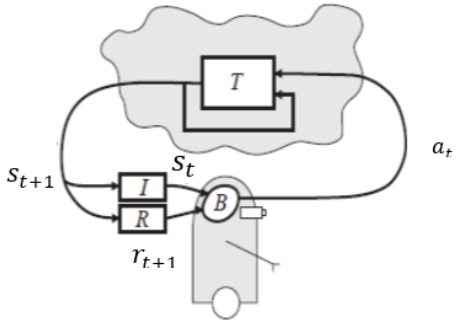


شکل ۱: مدل سیستم

۳- پیشینه نظری یادگیری تقویتی

از آنجا که پیش زمینه تئوری روش زمان‌بندی پیشنهادی مبتنی بر یادگیری تقویتی است، در این بخش به تشریح مسائل یادگیری تقویتی پرداخته شده است. یادگیری تقویتی یکی از انواع الگوریتم‌های یادگیری ماشین است، با این هدف که به عامل اجازه یادگیری می‌دهد که چگونه در محیط ناشناخته رفتار کند به طوری که در هر مرحله از میان مجموعه اعمال مجاز، عملکرد مناسب را با توجه به موقعیت و بازخوردی که از محیط می‌گیرد (این بازخورد شامل یک سیگنال پاداش یا جریمه اسکالر می‌باشد.) انتخاب کند. این چگونگی نگاشت موقعیت‌ها به عمل‌ها باید به گونه باشد که در بلندمدت پاداش بیشینه شود. در واقع عامل بدون آموزش و تنها با کسب تجربه در محیط پیرامونش راه رسیدن به هدف را می‌یابد.

یادگیری تقویتی عبارت است از یادگیری آنچه که عامل باید انجام دهد و چگونگی نگاشت حالت‌ها به اعمال به طوری که سیگنال عددی پاداش را در درازمدت به حداکثر رساند. برخلاف اکثر روش‌های یادگیری ماشین در این روش یادگیرنده هیچ دانش قبلی در مورد انتخاب عمل ندارد بلکه باید کشف کند با انتخاب کدام اعمال بیشترین پاداش را در طول زمان به دست می‌آورد. این اعمال نه تنها بر روی پاداش آنی که عامل از محیط می‌گیرد تأثیر می‌گذارند بلکه بر روی پاداش وضعیت بعدی و تمامی پاداش‌های بعد از آن هم تأثیرگذار است [۱۲]. علاوه بر عامل و محیط چهار عنصر اصلی سیستم یادگیری تقویتی شامل: سیاست ، تابع پاداش ، تابع ارزش و به صورت اختیاری مدلی از محیط است. در مدل استاندارد یادگیری تقویتی عامل از طریق عمل و ادراک در دنباله‌ای از گام‌های زمانی گسسته با محیط ارتباط برقرار می‌کند. همان‌طور که در شکل ۲ نشان داده شده است [۱۳]. عامل B یک یک تابع I دارد که وضعیت فعلی محیط را به عنوان ورودی s_t از محیط دریافت کرده و مشخص می‌کند که عامل چگونه وضعیت محیط را مشاهده نماید. وضعیت فعلی با $s_t \in S$ بیان می‌شود که S بیانگر مجموعه تمام حالات ممکن است. در این وضعیت عامل یک عمل $a_t \in A(s_t)$ از مجموعه عمل‌های ممکن در این حالت را برمی‌گزیند و با انتخاب عمل a_t به حالت بعدی s_{t+1} می‌رود و بلافاصله پاداش $r_{t+1} \in R$ را از محیط از طریق تابع R دریافت می‌کند. ارزش این انتقال حالت بستگی به سیگنال اسکالر r دارد. در کل عامل باید عملی را که منجر به افزایش سیگنال تقویت‌کننده باشد، انتخاب کند و این را در طول زمان با آزمایش و خطا یاد می‌گیرد. سیاست با $\pi: S \rightarrow A$ نشان داده می‌شود که π شامل نگاشتی از s به a است به طوریکه π_t (s,a) احتمال انتخاب عمل a در حالت s در گام زمانی t است. در مسائل مرحله‌ای حالت پایانی را می‌توان یک حالت جذب‌کننده تعریف کرد که بیانگر حالتی است که عامل در طول T گام زمانی به آن برخورد می‌کند بطوریکه با انجام هر عملی حالت خودش را حفظ می‌کند و پاداشی برابر صفر دریافت می‌کند.



شکل ۲: مدل استاندارد یادگیری تقویتی

ویژگی مارکف در یادگیری تقویتی اشاره به ویژگی بدون حافظه بودن در فرایندهای تصادفی دارد. وقتی حالت تمام اطلاعات لازم را برای انتخاب عمل در اختیار عامل بگذارد به‌گونه‌ای که عامل با قرار گرفتن در این حالت بدون نیاز به وقایع گذشته، خلاصه‌ای از گذشته را درک کند و تصمیمات آتی عامل فقط بستگی به وضعیت کنونی داشته باشد، این حالت دارای ویژگی مارکف می‌باشد. به طوریکه برای تمام s_t، a_t،r،s' داشته باشیم:

www.SID.ir

Archive of SID

$$P_r\{s_{t+1}=s', r_{t+1}=r|s_t,a_t\} \quad (۲)$$

در این صورت می‌توان حالت بعدی و پاداش مورد انتظار را با توجه به حالت جاری و عمل معین پیش‌بینی نمود. اگر مسئله یادگیری تقویتی دارای خاصیت مارکف باشد می‌توان آن را یک فرایند تصمیم‌گیری مارکف (MDP)[۲] در نظر گرفت. هدف MDP ایجاد یک سیاست مطلوب برای انتخاب عمل است به طوریکه بتواند تابع R_t که بیانگر امید ریاضی پاداش (مجموع پاداش‌ها با شروع از زمان t تا رسیدن به حالت نهایی) است را حداکثر کند.

$$R_t = r_{(t+1)} + \gamma r_{(t+2)} + \gamma^2 r_{(t+3)} + \ldots + \gamma^T r_{(t+T+1)} = \sum_{(k=0)}^{T} \gamma^k r_{t+(k+1)} \quad (۳)$$

γ فاکتور تخفیف یا کاهش است که میزان اهمیت پاداش‌های آینده را مشخص می‌کند و کمک به همگرایی مقدار تابع می‌کند. این فاکتور می تواند مقداری بین صفر تا یک اتخاذ کند. ($0<\gamma<1$) در فرایند تصمیم‌گیری مارکف با وجود هر حالت s و عمل a، احتمال انتقال به حالت 's برابر مقدار پاداشی است که دریافت می‌کند و با رابطه (۴) تعریف می‌شود.

$$P_a(s,s') = P_r(s_{t+1}=s'|s_t=s, a_t=a) \quad (۴)$$

همچنین مقدار مورد انتظار پاداش بعدی با رابطه (۵) بیان می‌شود.

$$R_a(s,s') = E(r_{t+1}|s_t=s, a_t=a, s_{t+1}=s') \quad (۵)$$

رابطه (۵) نشان‌دهنده عدم وابستگی پاداش مورد انتظار حالت بعد به حالت و اعمال قبلی است و فقط به حالت فعلی s و حالت بعدی 's بستگی دارد. با استفاده از دو رابطه (۴) و (۵)، محیط برای عامل شناسایی می‌شود.

۳-۱- روش یادگیری کیو

یادگیری کیو یکی از مهم‌ترین دستاوردهای یادگیری تقویتی می‌باشد و توسعه‌یافته روش یادگیری تفاضل موقتی است، که با توجه به عدم دسترسی به مدل کامل از محیط برای ایجاد سیاست مطلوب مورد استفاده قرار می‌گیرد. از آنجا که الگوریتم یادگیری کیو یک روش بدون مدل است و سعی در بیشینه‌سازی پاداش‌های جمع‌آوری شده در بلندمدت به جای افزایش پاداش‌های فوری دارد و می‌تواند خود را با یک محیط پویا و پیچیده وفق دهد، در روش پیشنهادی از روش یادگیری کیو استفاده شده است. این الگوریتم هر بار که با حالتی از محیط ملاقات می‌کند در صورتی که حالت پایانی نباشد، ارزش عمل a در حالت s را پس از هر گام زمانی تخمین می‌زند که با $Q^{\pi}(s,a)$ نشان داده می‌شود. این روش از سیاست شبه حریصانه برای تصمیم‌گیری در انتخاب اعمال در حالت خاص استفاده می‌کند، این موضوع در اکثر موارد باعث انتخاب عامل به عملی است که بیشترین پاداش را دریافت کرده باشد. ارزش عمل $Q^{\pi}(s,a)$ برای هر زوج حالت- عمل در یک جدول ذخیره می‌گردد. این جدول معمولاً با **Q** مشخص می‌شود. هدف عامل بیشینه‌سازی پاداش خود در بلندمدت است، اگر چه اغلب موارد منافع کوتاه‌مدت را برای دستیابی به منافع بلندمدت از دست می‌دهد [۱۳]. رابطه (۶) روش بروز رسانی ارزش عمل را برای الگوریتم یادگیری کیو نشان می‌دهد:

$$Q(s_t,a_t) = Q(s_t,a_t) + \alpha\left(r_{t+1} + \gamma \max_a Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)\right) \quad (۶)$$

۴- روش پیشنهادی

مسئله زمان‌بندی وظایف را می‌توان یک فرایند تصمیم‌گیری مارکوف با فضای حالت S، مجموعه اقدامات A و تابع پاداش فوری r(s,a) در نظر گرفت. در ادامه خصوصیات محیط روش زمان‌بندی پیشنهادی را مورد بررسی قرار می‌دهیم:

فضای حالت: هر حالت در مسئله زمان‌بندی را می‌توان با ظرفیت اشغال شده هر VM و مجموع طول وظایف اختصاص یافته به VM ها تعریف کرد. اگر S_k نشان‌دهنده ظرفیت اشغال شده بافر ماشین مجازی kام باشد و T_k بیانگر مجموع طول وظایف تخصیص یافته به ماشین مجازی kام باشد، در پلت فرم ابری فضای حالت را می‌توان به صورت برداری با توجه به رابطه (۷) تعریف کرد.

$$VM_k = (S_1, S_2, \ldots, S_k, T_1, T_2, \ldots, T_k) \quad (۷)$$

کاهش فضای حالت: در الگوریتم یادگیری کیو با اضافه کردن حالات و اعمال، فضای حالت به صورت نمایی رشد می‌کند که ذخیره‌سازی آن بسیار دشوار است و موجب پایین آمدن سرعت یادگیری و عدم همگرایی برنامه می‌شود. یکی از راه‌های کاهش فضای حالت استفاده از منطق فازی جهت گسسته‌سازی فضای حالت پیوسته است. در روش پیشنهادی به دلیل آنکه طول وظایف می‌تواند مقدارهای متفاوتی را اتخاذ کند و در نظر گرفتن همه آنها موجب عدم ملاقات برخی حالت‌ها توسط زمان‌بند می‌شود از منطق فازی جهت گسسته سازی فضای حالت پیوسته و کاهش فضای حالت برای سرعت بخشیدن به روند یادگیری استفاده شده است. از رابطه (۸) جهت گسسته‌سازی طول وظایف استفاده می‌کنیم، به طوریکه L_{T_i} بیانگر طول وظیفه i ام و Range بیانگر مقدار انتخابی برای تعیین محدوده طول وظایف است.

$$Length_i = \left[\frac{L_{T_i}}{\mathrm{Range}}\right] \quad (۸)$$

مجموعه اعمال: در هر حالت عامل می‌تواند یکی از ماشین‌های مجازی موجود را انتخاب کند و درخواست کاربر را به آن ماشین اختصاص دهد. به عنوان مثال اگر سه VM موجود باشد عامل می‌تواند یکی از VM های شماره یک تا سه را انتخاب کند در نتیجه فضای عمل مسئله یک فضای گسسته است. در اینجا حالت بعدی که عامل در آن قرار می‌گیرد با توجه به حالت فعلی و عملی که عامل انجام می‌دهد تعیین می‌گردد و همچنین مقدار پاداشی که عامل در ازای انتخاب VM در هر گام زمانی به دست می‌آورد متفاوت است و اگر تخمینی از پاداش در گذشته به دست آورده باشد در حال حاضر معتبر نیست و باید به طور مکرر بروز شود، پس با یک محیط قطعی ولی پویا روبه‌رو هستیم. هدف ما در این مسئله تخصیص مناسب وظایف به VMها است به طوری که میانگین زمان

www.SID.ir

Archive of SID

پاسخ را کاهش دهیم. برای دستیابی به این هدف عامل باید دنباله‌ای از اعمال را انجام داده تا به هدف برسد و خطا در انتخاب یکی از اعمال منجر به عدم دستیابی عامل به هدف می‌گردد. به همین دلیل مسئله مذکور یک مسئله انجمنی می‌باشد و از طرفی بخاطر اینکه انتخاب یک عمل توسط عامل بر روی دیگر اعمال تاثیرگذار است روش پیشنهادی یک مسئله زنجیره‌ای است.

پاداش فوری: بیانگر بازتابی است که عامل با توجه به حالتی که که در آن قرار دارد و عملی که انجام می‌دهد دریافت می‌کند تا نسبت به درستی عملش آگاهی کسب کند و در نهایت خود را به هدف نهایی‌اش برساند. پاداشی که عامل در زمان انتخاب عمل دریافت می‌کند با رابطه (۹) تعریف می‌شود.

$$r=\begin{cases}1 if\ T_k=min(T_k)\\ -1 if\ T_k=max(T_k)\\ 0 \qquad other\ wise\end{cases} \qquad (۹)$$

اگر وظیفه کاربر به ماشین مجازی‌ای اختصاص یابد که مجموع طول وظایف آن نسبت به سایر ماشین‌های موجود حداقل باشد، عامل پاداش مثبت یک دریافت می‌کند ولی اگر به ماشینی اختصاص یابد که مجموع طول وظایف آن حداکثر باشد، پاداش منفی یک را دریافت می‌کند و در بقیه موارد عامل پاداش صفر دریافت می‌کند. روند کار روش پیشنهادی این گونه است که ابتدا جدول **Q** با مقدار اولیه صفر مقداردهی می‌شود. در این روش فرض شده در ابتدا زمان‌بند هیچ دانشی در اختیار ندارد. تمام ماشین‌های مجازی همگن هستند و سایز بافر هر کدام ثابت است. اگر تعداد VM ها را برابر سه فرض کنیم. حالت اولیه (0,0,0,0,0,0)= VM_3 ملاقات می‌شود که سه پارامتر اول نشان‌دهنده ظرفیت اشغال شده بافر هر VM و سه پارامتر بعدی به ترتیب نشان‌دهنده مجموع طول وظایف موجود در هر VM است. زمان‌بند با مشاهده حالتی از محیط یکی از اقدامات موجود A={1,2,3} را با سیاست شبه حریصانه انتخاب می‌کند. (در ابتدا چون ارزش عمل برای تمام حالات صفر است به صورت تصادفی عمل می‌کند.) البته در روش پیشنهادی با ایجاد تغییراتی در انتخاب اپسیلون سعی شده کاوش در مراحل اولیه بیشتر باشد و به مرور الگوریتم بتواند حریصانه تصمیم بگیرد و با بهره‌گیری از تجربیات به دست آمده از یادگیری‌های قبلی عمل کند. روش انتخاب اپسیلون اینگونه است که ابتدا مقداری برای آن بین صفر تا یک در نظر گرفته می‌شود سپس با توجه به تعداد دورلازم برای همگرایی مسئله (حد آستانه) در هر اپیزود مقدار اپسیلون با رابطه (۱۰) تغییر می‌کند.

$$\varepsilon=\varepsilon-\frac{\varepsilon}{total\ cycle}\times cycle\ number \qquad (۱۰)$$

که منظور از total cycle تعداد دور لازم برای همگرایی و cycle number بیانگر شماره دور است. این روش تعیین اپسیلون باعث می‌شود به مرور زمان اپسیلون به صفر همگرا شود و الگوریتم حریص‌تر عمل کند.

با انتخاب هر کدام از عمل‌ها زمان‌بند وظیفه کاربر را به آن VM مربوطه تخصیص می‌دهد و با توجه به تابع پاداش، پاداش فوری را در ازای تخصیص این وظیفه به VM دریافت می‌کند. پاداش فوری این‌گونه حساب می‌شود که اگر وظیفه به صف VM ای منتقل شد که کمترین طول صف را از میان بقیه VM ها دارا باشد، پاداش مثبت یک دریافت می‌کند و اگر عکس این موضوع اتفاق بیفتد، پاداش منفی یک و در بقیه حالات پاداش صفر دریافت می‌کند. این تابع پاداش منجر به تخصیص هر وظیفه به صفی با کمترین مجموع طول می‌شود که باعث تخصیص مناسب وظایف به ماشین‌های مجازی و کاهش زمان انتظار وظایف در صف بافر هر VM و کاهش زمان پاسخ وظایف در بلندمدت می‌گردد. زمان‌بند پس از دریافت پاداش، تابع ارزش عمل را با رابطه (۶) بروز رسانی می‌کند سپس با تغییر فضای حالت اشغال شده بافر ماشین مجازی و مجموع طول وظایف، زمان‌بند حالت بعدی فضای حالت را مشاهده می کند. این روند تا پایان وظایف تکرار می‌شود و در پایان هر مرحله شرط همگرایی بررسی می‌شود. این روش تحت دو شرط: اکثر زوج حالت- عمل‌ها ملاقات شوند (برنامه تا یک حد آستانه‌ای مورد آزمایش قرار بگیرد.) و انتخاب بهترین اعمال نسبت به مرحله‌های قبلی هیچ تغییری نداشته باشند، قادر به همگرایی به تابع ارزش بهینه است. در صورت عدم همگرایی مراحل به‌صورت مکرر تکرار می‌شوند تا در نهایت زمان‌بند به یک مقدار یکتا همگرا شود تا بتوان به نتایج بدست آمده اعتماد کرد.

۴-۱- شبه کد روش پیشنهادی

با توجه به توضیحات گفته شده راجع به روش پیشنهادی، شبه کد روش پیشنهادی برای زمان‌بندی وظایف بر روی ماشین‌های مجازی با استفاده از الگوریتم یادگیری کیو در شکل ۳ نشان داده شده است.

1. Initialize *Q table*
2. best action and repeater initialize by 0
3. Repeat (for each episode)
    4. Initialize s_t
    5. Repeat (for each step of episode)
        6. a_t=actionSelection(s_t,Q) using ε-greedy policy
        7. Take action a_t observe r, s_{t+1}
        8. $Q(s_t,a_t)= Q(s_t,a_t)+ \alpha[r_{t+1}+\gamma \max_a Q(s_{t+1},a_{t+1})- Q(s_t,a_t)]$
        9. $s_t = s_{t+1}$
    10. end for until s_t is terminal
    11. repeater++;
    12. calculate best action and repeater
13. end for until best action not change and repeater > θ

شکل ۳: شبه کد روش پیشنهادی

توضیح شبه کد روش پیشنهادی با توجه به شکل ۳ به شرح زیر است.

۱) جدول **Q** به ازای تمام زوج حالت- عمل‌ها مقداردهی می‌شود. (معمولاً صفر)

۲) یک شمارشگر برای تعیین حد آستانه برای ملاقات تمام حالت‌ها به نام repeater و همچنین آرایه‌ای به نام best action جهت تعیین بهترین اعمال انتخاب شده در هر مرحله، تعریف می‌شود که در ابتدا با صفر مقداردهی می‌شوند. لازم به ذکر است این دو متغیر برای بررسی همگرایی مسئله به کار برده می‌شوند.

www.SID.ir

Archive of SID

۳) در هر اپیزود تکرار می‌شود.
۴) حالت فعلی فضای حالت S_t مشاهده می‌شود.
۵) تا ملاقات حالت پایانی تکرار می‌شود.
۶) با مشاهده حالت فعلی سیستم عمل a_t با سیاست شبه حریصانه انتخاب می‌شود.
۷)عمل a_t اجرا می‌شود، پاداشr وحالت بعدی سیستم S_{t+1} مشاهده می‌شود.
۸) مقدار $\mathbf{Q}$ به ازای حالت S_t و عمل a_t بروز می‌شود.
۹) حالت جدید S_{t+1} به عنوان حالت جدید سیستم در نظر گرفته می شود.
۱۰) مراحل ۶ تا ۹ را تا مشاهده حالت پایانی تکرار کن.
۱۱) repeater یک واحد اضافه می‌شود.
۱۲) best action وrepeater به جهت انتخاب بهترین اعمال و همچنین تعداد دور لازم برای ملاقات بیشتر حالت‌ها بررسی می‌شوند. (بررسی همگرایی مسئله)
۱۳) پایان برنامه تا زمانی که best action تغییر نکند و همچنین repeater از یک حد آستانه بیشتر باشد.

۵- ارزیابی نتایج

در این مقاله به منظور اعتبار سنجی و بازده روش پیشنهادی از شبیه‌ساز کلودسیم استفاده شده است. ابتدا همگرایی روش پیشنهادی بررسی می شود سپس دو سناریو مطرح شده و در هر سناریو زمان انتظار، زمان پاسخ و زمان تکمیل کار در روش پیشنهادی و سایر روش‌های زمان‌بندی کلاسیک مقایسه می‌شوند، در ادامه نرخ بهره‌وری در هر یک از سناریوها بررسی می‌شود. در آخر هم میزان تأثیر افزایش ظرفیت بافر در روش پیشنهادی و الگوریتم‌های زمان‌بندی مورد تحلیل قرار می‌گیرد. در این شبیه‌ساز الگوریتم‌های زمان‌بندی زیر برای مقایسه در نظر گرفته شده‌اند:
۱) روش زمان‌بندی پیشنهادی با Q-learning مشخص شده است که استفاده از روش یادگیری تقویتی منجر به زمان‌بندی بهینه وظایف بر روی ماشین‌های مجازی می‌گردد. در این روش وظایف با استفاده از روش شبه حریصانه طبق الگوریتم یادگیری کیو که در بخش قبلی اشاره شد، زمان‌بندی می‌شوند.
۲) روش زمان‌بندی تصادفی با Random مشخص شده است که وظایف کاربران را به صورت تصادفی بر روی ماشین‌های مجازی قرار می‌دهد. اگر ظرفیت باقی‌مانده بافر ماشین مجازی خالی باشد، وظایف دوباره زمان‌بندی می‌شوند و به صورت تصادفی بر روی دیگر ماشین‌های مجازی که ظرفیت آن‌ها هنوز پر نشده است، قرار می‌گیرند.
۳) روش زمان‌بندی ترتیبی با FIFo مشخص شده است. این روش ساده‌ترین روش زمان‌بندی است که وظایف کاربران را به ترتیب به ماشین‌های مجازی تخصیص می‌دهد. الگوریتم FIFO اگر وظایف به صورت صعودی وارد شوند بهترین عملکرد و اگر به صورت نزولی وارد شوند بدترین عملکرد را دارد.
۴) روش زمان‌بندی ترکیبی با Mix مشخص شده است که در ابتدا درخواست کاربران را به صورت تصادفی به ماشین‌های مجازی تخصیص می‌دهد ولی اگر ظرفیت باقی‌مانده بافر ماشین مجازی خالی بود، این وظایف دوباره زمان‌بندی می‌شوند و از بین ماشین‌های مجازی موجود به ماشین مجازی با بیشترین ظرفیت خالی بافر تخصیص می‌یابند.
۵) روش زمان‌بندی حریصانه با Greedy مشخص شده است که با ورود هر درخواست آن را از بین ماشین‌ها به بهترین ماشین مجازی از نظر ظرفیت باقی‌مانده بافر تخصیص می‌دهد. این روش با وجود تفاوت‌های که با روش FIFo دارد ولی از آنجایی که وظایف را به ترتیب به ماشین‌ها بر اساس ظرفیت اشغال شده کمتر تخصیص می‌دهد تقریباً مشابه این روش است.
۶) روش زمان‌بندی مرجع [۲] که با Q-sch مشخص شده است. در روش Q-sch از یادگیری کیو برای زمان‌بندی وظایف بر روی منابع استفاده شده است که در آن فضای حالت را فقط ظرفیت باقی مانده بافر ماشین‌های مجازی در نظر می‌گیرد، در ابتدا تمامی صف بافرها خالی است و هر بار که وظیفه‌ای به ماشین مجازی نگاشت می‌شود از ظرفیت بافر کم می‌شود تا زمانی که همه وظایف نگاشت شوند. تابع پاداش در این روش به این صورت است که ارسال کننده وظایف هنگامی پاداش مثبت یک از محیط دریافت می‌کند، که هم ظرفیت بافر باقی آن ماشین مجازی نسبت به دیگر ماشین‌ها حداکثر باشد و هم میانگین زمان انتظار صف بافر آن ماشین مجازی کمتر باشد و اگر تنها شرط میانگین زمان انتظار برقرار باشد، پاداش صفر و در غیر این صورت پاداش منفی یک دریافت می‌کند و در نهایت تابع حالت – عمل را بروز رسانی می‌کند تا زمانی که خطا کمتر از یک حد آستانه باشد. (خطا در این روش از تفاضل مقدار فعلی ارزش حالت – عمل با مقدار قبلی آن حاصل می‌شود.)

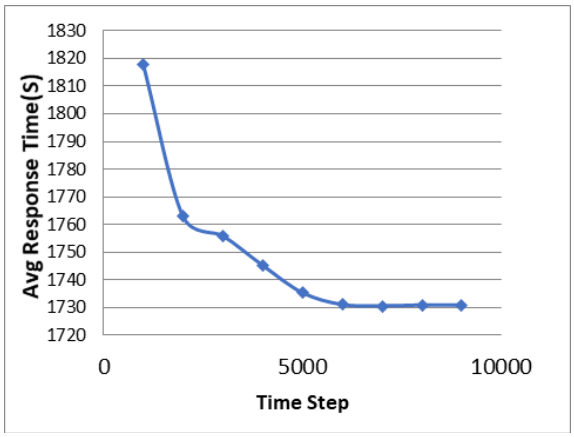
۵-۱- بررسی همگرایی الگوریتم پیشنهادی

در روش پیشنهادی زمان‌بند با ارسال وظایف به ماشین‌ها در هر مرحله از یادگیری کسب تجربه می‌کند تا زمانی که الگوریتم به یک مقدار یکتا همگرا شود. در هر دور زمان‌بند یکسری فضای حالت را مشاهده می‌کند و تابع ارزش را به ازای هر حالت تا زمانی که به حالت پایانی برسد، محاسبه می‌کند سپس تابع ارزش به ازای تمام حالت‌ها بروز رسانی می شود. پس از ملاقات حالت پایانی دوباره تمام عملیات به منظور محاسبه میزان تغییرات تابع ارزش در تمام حالات انجام می‌شوند و این بروز رسانی تا زمانی که الگوریتم به اندازه کافی فضای حالت‌ها را بازدید کند و به یک مقدار یکتا همگرا شود ادامه پیدا می‌کند. در اینجا میانگین زمان پاسخ را برای ۵۰ وظیفه محاسبه کرده‌ایم. با توجه به شکل ۴ مشاهده می‌شود هنگامی که تعداد دور بین ۱۰۰۰ تا ۴۰۰۰ باشد الگوریتم جواب درستی برنمی‌گرداند و میانگین زمان پاسخ حدود۱۸۲۰ ثانیه است ولی از ۵۰۰۰ دور به بعد الگوریتم به سمت جواب بهینه همگرا شده است و میانگین زمان پاسخ به ۱۷۳۰ ثانیه رسیده است و دیگر تغییر محسوسی نداشته است و می‌توان به جواب الگوریتم اعتماد کرد.

www.SID.ir

Archive of SID

در روش پیشنهادی علاوه بر تعداد دور لازم برای ملاقات بیشتر فضای حالت‌ها شرط دیگر، مقایسه بهترین عمل‌ها پس از پایان هر دور اضافه شده است. این شرط پس از پایان هر دور محاسبه می‌کند آیا بهترین اعمال انتخاب شده در این دور از یادگیری با مرحله قبلی باهم برابر هستند یا خیر؟ البته لازم به ذکر است که این شرط در کنار شرط تعداد دور لازم استفاده می‌شود. یعنی هم باید تعداد دور از یک حد آستانه بیشتر باشد مثلاً از ۵۰۰۰ دور بیشتر باشد و هم بهترین اعمال انتخابی با مرحله قبلی یکسان باشند. با توجه به نتایج به‌دست‌آمده پس از ۶۰۰۰ دور ما به جوابی نزدیک به جواب الگوریتم روش پیشنهادی می‌رسیم و این نشان می دهد که الگوریتم به یک مقدار یکتا همگرا شده است.



شکل۴: بررسی همگرایی روش پیشنهادی

۵-۲- سناریوهای ارزیابی

در این مقاله دو سناریو برای بررسی روش کار انجام می‌گیرد. در سناریوی اول نرخ ورود وظایف در محدوده ۱۰ تا ۲۰ وظیفه است، مشخصات پارامترهای شبیه‌سازی این سناریو در جدول ۱ آمده است. در سناریو دوم نرخ ورود وظایف در محدوده ۲۰ تا ۱۰۰ وظیفه است. در سناریو دوم اکثر وظایف طول بسیار کمتری نسبت به سناریو اول دارند، مشخصات آن در جدول ۲ نشان داده شده است. در شبیه‌سازی از یک یا دو میزبان مستقل استفاده شده است که شامل تعدادی ماشین مجازی با اندازه بافر ثابت هستند. ماشین‌های مجازی همگن در نظر گرفته شده‌اند و وظایف بر اساس یک دیتاست واقعی وارد سیستم می‌گردند. محدوده انتخابی وظایف به صورت تصادفی و به گونه‌ای است که وظایف از نظر طول دارای پراکندگی باشند. در هر سناریو به منظور تحلیل منصفانه، هر روش زمان‌بندی چندین بار اجرا شده و از نتایج به دست آمده میانگین گرفته شده است.

در روش‌های زمان‌بندی فاکتورهای مهمی مثل زمان پاسخ، زمان تکمیل کار، زمان انتظار، زمان اجرا، هزینه‌ها، نرخ بهره‌وری، توازن بار و دیگر موارد اهمیت دارد. در این مقاله ما فاکتورهای مهم زمان پاسخ، زمان انتظار و زمان تکمیل کار را مورد توجه قرار داده‌ایم. همچنین نرخ بهره‌وری و توزیع بارکاری را نیز بررسی می‌کنیم. زمان پاسخ مدت زمانی است که از ارسال یک وظیفه تا اتمام آن وظیفه طول می‌کشد. زمان انتظار بیانگر زمان انتظار وظایف در صف بافر ماشین‌های مجازی تا زمان شروع اجرا است. زمان تکمیل کار مدت زمانی است که کل وظایف اجرا می‌شوند و با زمان اتمام آخرین وظیفه محاسبه می‌شود. برای محاسبه زمان پاسخ و زمان انتظار و زمان تکمیل کار به ترتیب از رابطه‌های ۱۱ و ۱۲ و ۱۳ استفاده می‌کنیم به طوری که F_{T_i} بیانگر زمان اتمام کار، S_{T_i} بیانگر زمان ارسال کار، E_{T_i} بیانگر زمان اجرا وظایف و M_{T_i} بیانگر زمان اتمام وظایف می‌باشند.N تعداد کل وظایف و Tasks مجموعه وظایف ورودی است.

$$Average\ Response Time = \frac{1}{N}\sum_{i\in Tasks}(F_{T_i} - S_{T_i}) \quad (۱۱)$$

$$Average Wating\ Time = \frac{1}{N}\sum_{i\in Tasks}(F_{T_i} - S_{T_i} - E_{T_i}) \quad (۱۲)$$

$$Makespan = max_{i\in Tasks} M_{T_i} \quad (۱۳)$$

جدول ۱: پارامترهای سناریوی اول

parameters	values
Length of job	5000-200000 million Instructions
Total number of jobs	20
Total number of VMs	3
VM frequency	1000 million Instructions per second
VM memory(RAM)	1740 mega byte
VM bandwidth	1000 mega byte per second
Number of VM buffer	5-15
Number of PEs	1
Number of Datacenters	1
Number of hosts	1

جدول ۲: پارامترهای سناریوی دوم

parameters	values
Length of job	100-400000 million Instructions
Total number of jobs	100
Total number of VMs	3
VM frequency	1000 million Instructions per second
VM memory(RAM)	1740 mega byte
VM bandwidth	1000 mega byte per Second
Number of VM buffer	5-50
Number of PEs	5
Number of Datacenters	1
Number of hosts	2

۵-۳- میانگین زمان تکمیل کار

زمان تکمیل کار (Makespan) حداکثر اختلاف زمانی بین شروع و پایان دنباله‌ای از وظایف پس از تکمیل آخرین وظیفه است. در اکثر الگوریتم‌ها از دید مدیر سیستم یکی از عامل‌های مهم در زمان‌بندی بهینه، زمان تکمیل کار کوچک است زیرا هرچه کمتر باشد کیفیت خدمات بیشتر می‌شود و همچنین مدیر، کاربران بیشتری را سرویس‌دهی می‌کند. روش پیشنهادی با ایجاد نگاشت مناسبی از وظایف به منابع باعث می‌شود وظایف با طول کمتر، منتظر وظایف با طول بیشتر نشوند که این موضوع منجر به کاهش زمان تکمیل کار می‌شود. با توجه به نتایج به دست آمده از جدول ۳، در سناریو اول در تمام مراحل، الگوریتم پیشنهادی زمان تکمیل کار کمتری نسبت به دیگر روش‌های زمان‌بندی داشته است با

www.SID.ir

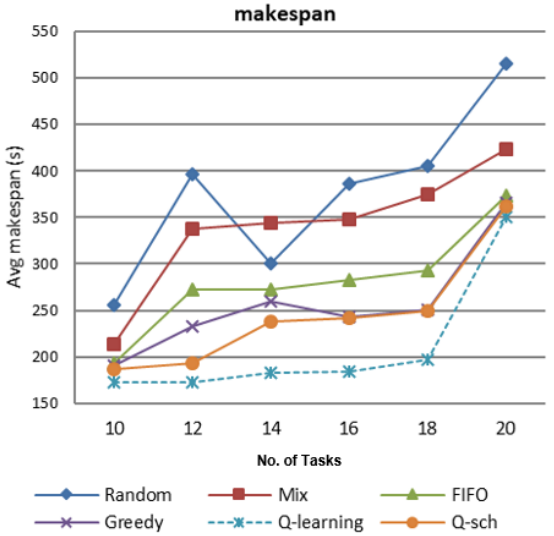
Archive of SID

افزایش تعداد وظایف، زمان تکمیل کار افزایش یافته است. این الگوریتم حتی در مراحلی که چینش وظایف به‌گونه‌ای باشد که الگوریتم FIFO و Greedy به صورت تصادفی عملکرد خوبی داشته باشند (وظایف از نظر طول صعودی وارد سیستم گردند) به عنوان مثال زمانی که تعداد وظایف برابر ۱۰ یا ۲۰ باشد، باز هم زمان تکمیل کار را تا جای ممکن کاهش می‌دهد که نسبتاً میزان کاهش آن کمتر است. ولی در دیگر مراحل کاهش قابل توجهی داشته است. روش پیشنهادی توانسته است زمان تکمیل کار را نسبت به روش Q-sch نیز بهبود ببخشد به طوریکه بیشترین بهبود را با ۱۶ وظیفه با اختلاف زمان ۵۷ ثانیه نسبت به روش Q-sch به دست آورده است.

جدول ۳: میانگین زمان تکمیل کار سناریو اول(ثانیه)

تعداد وظایف الگوریتم	۲۰	۴۰	۶۰	۸۰	۱۰۰
Random	۴۵۰	۶۳۵	۷۷۵	۱۳۱۳	۱۱۱۴
Mix	۴۴۶	۶۰۳	۶۷۷	۱۰۹۷	۱۰۹۷
FIFO	۴۴۳	۵۳۹	۶۹۸	۱۰۷۳	۱۰۸۶
Greedy	۴۴۱	۵۴۲	۶۴۶	۱۰۵۹	۹۶۶
Q-learning	۴۳۲	۴۳۲	۴۶۶	۸۲۰	۸۲۱
Q-sch	۴۳۷	۵۳۷	۵۸۵	۹۵۷	۸۹۷

در جدول ۴ اختلاف نتایج الگوریتم‌های مختلف با روش پیشنهادی در سناریو اول مقایسه شده است و درصد بهبود محاسبه گردیده است. با توجه به نتایج به دست آمده زمان تکمیل کار در روش پیشنهادی در تمام مراحل بهتر از دیگر روش‌های زمان‌بندی بوده است. روش پیشنهادی بیشترین بهبود را در حدود ۴۰ تا ۵۰ درصد نسبت به دو الگوریتم Random و Mix و در حدود ۲۰ الی ۳۰ درصد نسبت به دو الگوریتم FIFO و Greedy و نسبت به روش Q-sch در حدود ۲۰ درصد بیشترین بهبود را داشته است. بیان این نکته نیز حائز اهمیت است که با اینکه الگوریتم پیشنهادی در بعضی موارد بهبود کمتری نسبت به مراحل دیگر داشته است ولی در عین حال مناسب‌ترین تخصیص را داشته است و نمی‌توان تحت این شرایط بهبود بیشتری را به دست آورد.



شکل ۵: میانگین زمان تکمیل کار در سناریوی اول

جدول ۴: بهبود میانگین زمان تکمیل کار روش پیشنهادی در سناریوی اول

تعداد وظایف / الگوریتم	۱۰	۱۲	۱۴	۱۶	۱۸	۲۰
Random	۳۲/۴۲	۵۶/۳۱	۳۹	۵۲/۳۳	۵۱/۶۰	۳۱/۹۷
Mix	۱۸/۷۷	۴۱/۸۱	۴۶/۸۰	۴۶/۹۷	۴۷/۵۹	۱۷/۲۱
FIFO	۱۰/۳۶	۳۶/۶۳	۳۲/۹۶	۳۴/۹۸	۳۳/۱۰	۶/۱۴
Greedy	۸/۹۴	۲۵/۷۵	۲۹/۶۱	۲۴/۲۷	۲۱/۹۱	۴/۰۹
Q-sch	۶/۹۸	۱۰/۳۶	۲۳/۱۰	۲۳/۶۵	۲۱/۲۸	۳/۰۳

الگوریتم پیشنهادی در سناریو دوم با وجود تعداد وظایف بیشتر نسبت به سناریو اول، باز هم کمترین زمان تکمیل کار را در مقایسه با سایر الگوریتم‌ها داراست. در این سناریو اکثر وظایف طول بسیار کمتری نسبت به سناریوی اول دارند به طوری که مثلاً با توجه به جدول ۵ زمانی که تعداد وظایف برابر ۲۰ است، الگوریتم پیشنهادی میانگین زمان تکمیل کارش تقریباً ۴۳۲ ثانیه است و الگوریتم Random با توجه به اینکه به طور میانگین نسبت به دیگر الگوریتم‌ها بدترین تخصیص وظایف را انجام می‌دهد، میانگین زمان تکمیل کارش تقریباً ۴۵۰ ثانیه است. پس می‌توان نتیجه گرفت طول وظایف بسیار کوچک هستند و همین مقدار کاهش بسیار قابل توجه است. با توجه به شکل ۵ روش پیشنهادی توانسته با افزایش تعداد وظایف هم میانگین زمان تکمیل کار را نسبت به تعداد وظایف کم، بهبود ببخشد. پس می‌توان از روش پیشنهادی در تعداد وظایف بیشتری نیز استفاده کرد.

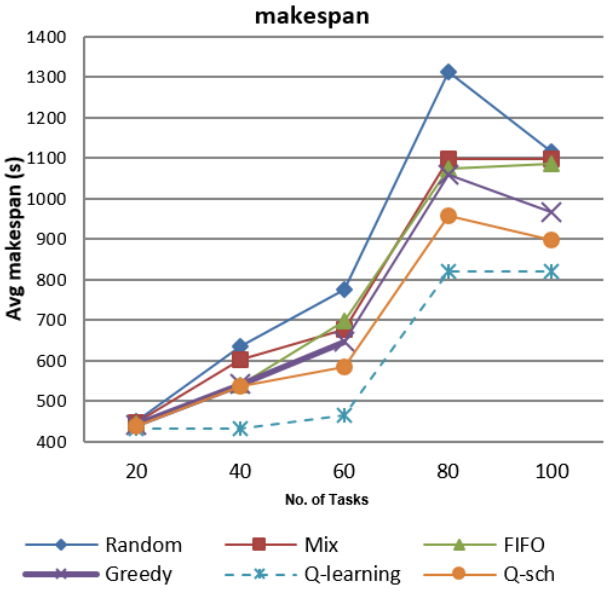
جدول ۵: میانگین زمان تکمیل کار سناریو دوم(ثانیه)

تعداد وظایف الگوریتم	۲۰	۴۰	۶۰	۸۰	۱۰۰
Random	۴۵۰	۶۳۵	۷۷۵	۱۳۱۳	۱۱۱۴
Mix	۴۴۶	۶۰۳	۶۷۷	۱۰۹۷	۱۰۹۷
FIFO	۴۴۳	۵۳۹	۶۹۸	۱۰۷۳	۱۰۸۶
Greedy	۴۴۱	۵۴۲	۶۴۶	۱۰۵۹	۹۶۶
Q-learning	۴۳۲	۴۳۲	۴۶۶	۸۲۰	۸۲۱
Q-sch	۴۳۷	۵۳۷	۵۸۵	۹۵۷	۸۹۷

طبق جدول ۶ می‌توان استدلال کرد که روش پیشنهادی نه تنها برای تعداد وظایف کم بلکه برای تعداد وظایف زیاد هم میانگین زمان تکمیل را بهبود می‌بخشد بطوریکه روش پیشنهادی در محدوده ۱۰۰ وظیفه توانسته است نسبت به روش Greedy حدود ۱۵ درصد، نسبت به روش FIFO حدود ۲۴ درصد، نسبت به روش Mix و Random به ترتیب ۲۵ و ۲۶ درصد و نسبت به روش Q-sch حدود ۸ درصد میانگین زمان تکمیل‌کار را بهبود ببخشد. طبق جدول ۶ روش پیشنهادی کمترین درصد بهبود میانگین زمان تکمیل کار را نسبت به الگوریتم Q-sch با اختلاف ۱،۱۴ درصد در محدوده ۲۰ وظیفه داشته است.

www.SID.ir

Archive of SID



شکل ۶: میانگین زمان تکمیل کار سناریو دوم

جدول ۶: بهبود میانگین زمان تکمیل کار روش پیشنهادی در سناریوی دوم (درصد)

تعداد وظایف / الگوریتم	۲۰	۴۰	۶۰	۸۰	۱۰۰
Random	۴	۳۱/۹۶	۳۹/۸۷	۳۷/۵۴	۲۶/۳۰
Mix	۳/۱۳	۲۸/۳۵	۳۱/۱۶	۲۵/۲۵	۲۵/۱۵
FIFO	۲/۴۸	۱۹/۸۵	۳۳/۲۳	۲۳/۵۷	۲۴/۴۰
Greedy	۲/۰۴	۲۰/۲۹	۲۷/۸۶	۲۲/۵۶	۱۵/۰۱
Q-sch	۱/۱۴	۱۹/۵۵	۲۰/۳۴	۱۴/۳۱	۸/۴۷

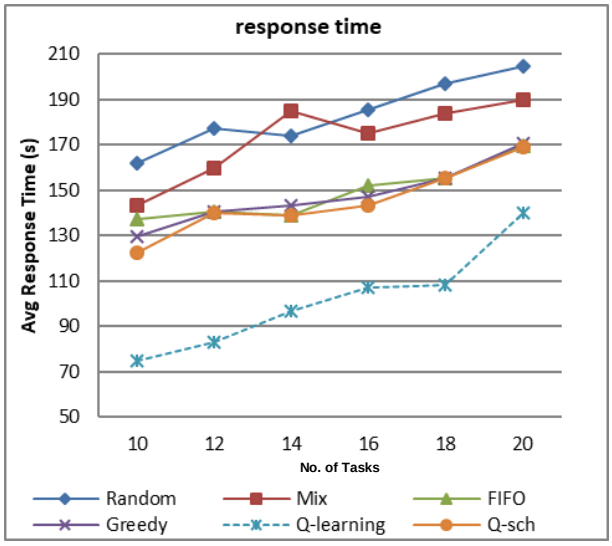
۵-۴- میانگین زمان پاسخ

زمان پاسخ برابر کل زمان لازم برای پاسخ به درخواست کاربران است. همچنین می‌توان زمان پاسخ را با مجموع زمان سرویس و زمان انتظار یک وظیفه نیز تعریف کرد. زمان پاسخ از آنجا که در رضایت‌مندی کاربر از سرویس تأثیر بسزایی دارد، یکی از فاکتورهای اصلی کیفیت سرویس است. با توجه به اینکه روش زمان‌بندی پیشنهادی طول وظایف جاری روی ماشین‌های مجازی را در نظر می‌گیرد می‌تواند راه‌حل بهینه را برای به حداقل رساندن زمان پاسخ پیدا کند. نتایج میانگین زمان پاسخ سناریو اول در جدول ۷ نشان داده شده است. همانطور که مشخص است، زمان پاسخ در هر روش زمان‌بندی با افزایش نرخ ورود افزایش یافته است. روش زمان‌بندی پیشنهادی زمان پاسخ را به طور مؤثری نسبت به سایر روش‌ها کاهش داده است. زمان‌بندی Random با تخصیص تصادفی وظایف بر روی ماشین‌ها، بدون در نظر گرفتن حجم کاری ماشین‌ها طولانی‌ترین زمان پاسخ را ایجاد می‌کند. روش زمان‌بندی Mix در اکثر مواقع روش زمان‌بندی Random را بهبود می‌بخشد. الگوریتم FIFO با توجه به پراکندگی وظایف از نظر طول، عملکرد خوبی در زمان پاسخ ندارد. الگوریتم Greedy نیز رویکرد مشابهی به الگوریتم FIFO دارد. در روش Q-sch با توجه به اینکه فضای حالت را تنها ظرفیت باقیمانده بافر ماشین‌های مجازی در نظر می‌گیرد نمی‌تواند به عامل درکی صحیحی از محیط بدهد همچنین در این روش پاداش دهی را بر اساس میانگین زمان انتظار وظایف و ظرفیت بافر انجام می‌دهد که لزوماً منجر به تخصیص مناسب وظایف نمی‌گردد ولی در روش پیشنهادی با گسترش فضای حالت و تغییر در روش پاداش دهی، در میانگین زمان پاسخ نسبت به روش Q-sch بهبود حاصل شده است.

جدول۷: میانگین زمان پاسخ سناریو اول(ثانیه)

تعداد وظایف / الگوریتم	۱۰	۱۲	۱۴	۱۶	۱۸	۲۰
Random	۱۶۲	۷۷	۱۷۴	۱۸۶	۱۹۷	۲۰۵
Mix	۱۴۳	۱۶۰	۱۸۵	۱۷۵	۱۸۴	۱۹۰
FIFO	۱۳۷	۱۴۱	۱۳۹	۱۵۲	۱۵۵	۱۷۰
Greedy	۱۳۰	۱۴۱	۱۴۳	۱۴۷	۱۵۵	۱۷۱
Q-learning	۷۵	۸۳	۹۷	۱۰۷	۱۰۸	۱۴۰
Q-sch	۱۲۲	۱۴۰	۱۳۹	۱۴۳	۱۵۵	۱۶۹

در روش پیشنهادی در تعیین فضای حالت علاوه بر ظرفیت بافر ماشینهای مجازی، مجموع طول وظایف در صف بافر هر ماشین مجازی نیز در نظر گرفته می‌شود که برای جلوگیری از افزایش فضای حالت از منطق فازی در تعیین محدوده طول وظایف استفاده شده است، همچنین پاداش دهی براساس مجموع طول صف بافر هر ماشین مقداردهی شده است. نتایج در جدول ۷ نشان می‌دهد که روش پیشنهادی در مقایسه با Q-sch از نظر میانگین زمان پاسخ بهبود یافته است. روش Q-sch توانسته زمان پاسخ را نسبت به روشهای FIFO و Greedy بهبود دهد ولی این بهبود قابل توجه نیست.



شکل ۷: میانگین زمان پاسخ سناریوی اول

با توجه به نتایج بدست آمده در جدول ۸ روش پیشنهادی به طور میانگین نسبت به روش Random حدود ۴۵ درصد، نسبت به روش Mix حدود ۴۰ درصد، نسبت به روش FIFO و Greedy به ترتیب ۳۲ و ۳۱ درصد و همچنین نسبت به روش Q-sch حدود ۳۰ درصد بهبود داشته است. بیشترین بهبود را نسبت به روش Q-sch با ۱۲ وظیفه در حدود

www.SID.ir

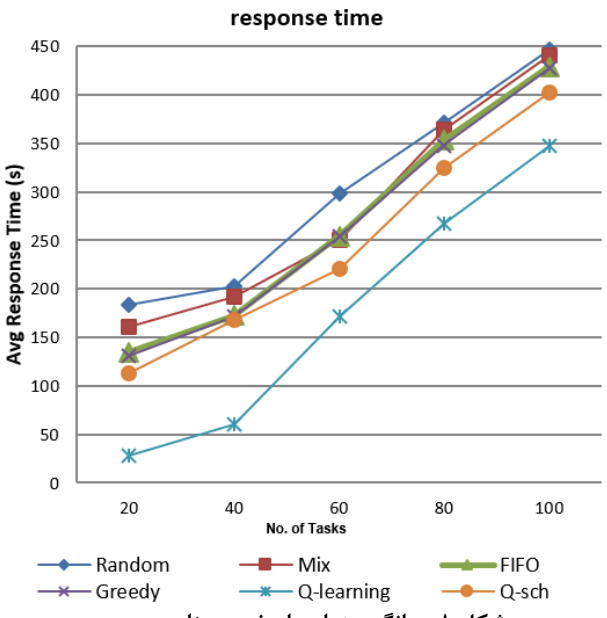
Archive of SID

۴۰ درصد و کمترین بهبود را با ۲۰ وظیفه حدود ۱۷ درصد داشته است. در نتیجه در تمام مراحل روش پیشنهادی کاهش زمان پاسخ قابل توجهی نسبت به سایر الگوریتم‌ها در سناریو اول داشته است.

جدول ۸ : بهبود میانگین زمان پاسخ روش پیشنهادی در سناریو اول(درصد)

تعداد وظایف / الگوریتم	۱۰	۱۲	۱۴	۱۶	۱۸	۲۰
Random	۵۳/۷۰	۵۳/۱۰	۴۵/۲۲	۴۲/۴۷	۴۵/۱۷	۳۱/۷۰
Mix	۴۷/۵۵	۴۸/۱۲	۴۷/۵۶	۳۸/۸۵	۴۱/۳۰	۲۶/۳۱
FIFO	۴۵/۲۵	۴۱/۱۳	۳۰/۲۱	۲۹/۶۰	۳۰/۳۲	۱۷/۶۴
Greedy	۴۲/۳۰	۴۱/۱۳	۳۲/۱۶	۲۷/۲۱	۳۰/۳۲	۱۸/۱۲
Q-sch	۳۸/۵۲	۴۰/۱۷	۳۰/۲۱	۲۵/۱۷	۳۰/۳۲	۱۷/۱۵

در سناریو دوم با توجه به افزایش تعداد وظایف، الگوریتم پیشنهادی بهترین عملکرد را در میان سایر الگوریتم‌ها داشته است. همان‌طور که در جدول ۹ قابل مشاهده است با افزایش تعداد وظایف، زمان پاسخ هم افزایش یافته است ولی در روش پیشنهادی این افزایش با شیب کمی همراه است. با اینکه روش پیشنهادی در سناریو دوم در زمان تکمیل کار به خاطر کوچک بودن طول وظایف نتوانسته بود اختلاف بالایی با دیگر الگوریتم‌ها ایجاد کند ولی در زمان پاسخ این اختلاف قابل توجه است زیرا درست است که اکثر وظایف کوچک هستند ولی در بین این وظایف کوچک، وظیفه‌ای با طول بالا وارد سیستم می‌شود و تخصیص نامناسب منجر به افزایش بسزایی در زمان پاسخ دیگر وظایف می‌شود از این رو الگوریتم پیشنهادی با تجاربی که در طول مدت یادگیری به دست می آورد، از وقوع این اتفاق جلوگیری می‌کند و اجازه نمی‌دهد وظایف کوچک فدای وظایفی با طول بالا شوند.



شکل ۸: میانگین زمان پاسخ در سناریوی دوم

جدول ۹: میانگین زمان پاسخ در سناریو دوم (ثانیه)

تعداد وظایف / الگوریتم	۲۰	۴۰	۶۰	۸۰	۱۰۰
Random	۱۸۳	۲۰۳	۲۹۸	۳۷۱	۴۴۶
Mix	۱۶۱	۱۹۱	۲۵۰	۳۶۴	۴۴۰
FIFO	۱۳۴	۱۷۲	۲۵۴	۳۵۳	۴۲۹
Greedy	۱۳۰	۱۷۱	۲۵۴	۳۴۸	۴۲۷
Q-learning	۲۸	۶۰	۱۷۲	۲۶۷	۳۴۷
Q-sch	۱۱۳	۱۶۸	۲۲۱	۳۲۵	۴۰۳

همان‌طور که در جدول ۱۰ نشان داده شده است. روش پیشنهادی در سناریو دوم توانسته است در برخی مراحل در بهترین حالت حدود ۷۰ الی ۸۰ درصد نسبت به دیگر الگوریتم‌ها میانگین زمان پاسخ را بهبود دهد. کمترین بهبود هم مربوط به زمانی است که تعداد وظایف برابر ۱۰۰ هستند و بهبودی در حدود ۱۳ درصد نسبت به الگوریتم Q-sch داشته است که این درصد بهبود تحت آن شرایط آزمایش بسیار مؤثر است زیرا الگوریتم پیشنهادی تا حد ممکن توانسته میانگین زمان پاسخ را کاهش دهد.)

طبق جدول ۱۰ در تمام مراحل آزمایش در بین الگوریتم‌های رقیب، الگوریتم Random بدترین و الگوریتم Q-sch بهترین عملکرد را نسبت به سایر روش‌ها داشته‌اند.

جدول۱۰: بهبود میانگین زمان پاسخ روش پیشنهادی در سناریو دوم (درصد)

تعداد وظایف / الگوریتم	۲۰	۴۰	۶۰	۸۰	۱۰۰
Random	۸۴/۶۹	۷۰/۴۴	۴۲/۲۸	۲۸/۰۳	۲۲/۱۹
Mix	۸۲/۶۰	۶۸/۵۸	۳۱/۲۰	۲۶/۶۴	۲۱/۱۳
FIFO	۷۹/۱۰	۶۵/۱۱	۳۲/۲۸	۲۴/۳۶	۱۹/۱۱
Greedy	۷۸/۴۶	۶۴/۹۱	۳۲/۲۸	۲۳/۲۷	۱۸/۷۳
Q-sch	۷۵/۲۲	۶۴/۲۸	۲۲/۱۷	۱۷/۸۴	۱۳/۸۹

۵-۵- میانگین زمان انتظار

زمان انتظار مدت زمانی است که طول می‌کشد تا وظیفه در صف هر ماشین مجازی که به آن اختصاص یافته منتظر اجرا شود. اکثر الگوریتم‌های زمان‌بندی سعی در کاهش زمان انتظار در جهت بهبود کیفیت سرویس و افزایش رضایت‌مندی مشتری دارند. برای دستیابی به زمان پاسخ مطلوب باید زمان انتظار کمی داشته باشیم. همان‌طور که در جدول ۱۱ و شکل ۹ قابل مشاهده است، الگوریتم پیشنهادی از آنجا که وظایف کوتاه را منتظر وظایف طولانی نمی‌گذارد و هر وظیفه را به صفی با کمترین میزان زمان اجرا تخصیص می‌دهد باعث کاهش زمان انتظار

www.SID.ir

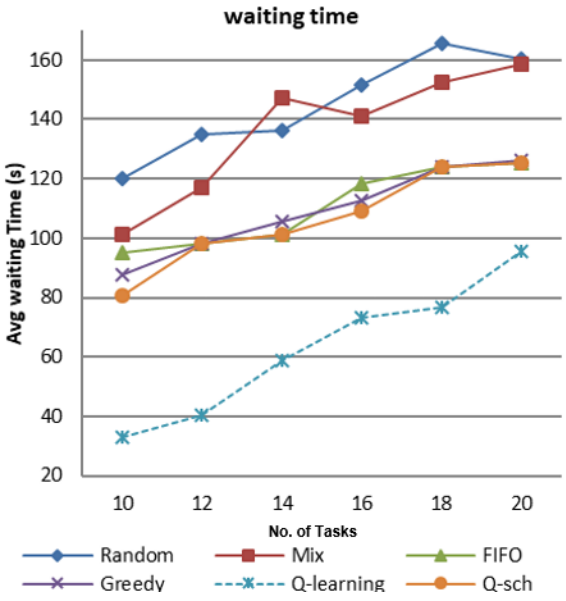
Archive of SID

وظایف در صف بافر ماشین‌های مجازی می‌گردد. روش پیشنهادی با کاهش حداکثری میانگین زمان انتظار توانسته است بهترین عملکرد را نسبت به سایر روش‌های زمان‌بندی موجود داشته باشد. در نتیجه در هر دوره حداقل زمان انتظار را نسبت به سایر روش‌های زمان‌بندی دارد و این روند را با افزایش تعداد وظایف نیز حفظ می‌کند.

جدول۱۱: میانگین زمان انتظار سناریوی اول (ثانیه)

تعداد وظایف / الگوریتم	۱۰	۱۲	۱۴	۱۶	۱۸	۲۰
Random	۱۲۰	۱۳۵	۱۳۶	۱۵۱	۱۶۶	۱۶۰
Mix	۱۰۱	۱۱۷	۱۴۷	۱۴۱	۱۵۳	۱۵۸
FIFO	۹۵	۹۸	۱۰۱	۱۱۸	۱۲۴	۱۲۵
Greedy	۸۸	۹۸	۱۰۶	۱۱۳	۱۲۴	۱۲۶
Q-learning	۳۳	۴۰	۵۹	۷۳	۷۷	۹۵
Q-sch	۸۰	۹۸	۱۰۱	۱۰۹	۱۲۳	۱۲۵

با توجه به جدول۱۲ در سناریو اول، روش پیشنهادی نسبت به الگوریتم‌های Random و Mix در بهترین حالت در حدود ۷۰ درصد و در بدترین حالت در حدود ۴۰ درصد بهبود داشته است، همچنین نسبت به روش FIFO و Greedy بهبودی در حدود ۲۰ الی ۶۰ درصد داشته است. روش پیشنهادی نسبت به روش Q-sch از نظر میانگین زمان پاسخ با تعداد ۱۰ وظیفه ۵۸ درصد، با تعداد ۱۲ وظیفه ۵۹ درصد و به ترتیب با ۱۴،۱۶،۱۸،۲۰ وظیفه به ترتیب ۴۱، ۳۳، ۳۷ و ۲۴ درصد میانگین زمان انتظار را بهبود بخشیده است. همان‌طور که قابل مشاهده است، روش پیشنهادی بیشترین بهبود را در میانگین زمان انتظار نسبت به میانگین زمان پاسخ و زمان تکمیل کار داشته است.



شکل ۹ : میانگین زمان انتظار سناریو اول

جدول۱۲: بهبود میانگین زمان انتظار روش پیشنهادی در سناریوی اول (درصد)

تعدادوظایف / الگوریتم	۱۰	۱۲	۱۴	۱۶	۱۸	۲۰
Random	۷۲/۵۰	۷۰/۳۷	۵۶/۶۱	۵۱/۶۵	۵۳/۶۱	۴۰/۶۲
Mix	۶۷/۳۲	۶۵/۸۱	۵۹/۸۶	۴۸/۲۲	۴۹/۶۷	۳۹/۸۷
Fifo	۶۵/۲۶	۵۹/۱۸	۴۱/۵۸	۳۸/۱۳	۳۷/۹۰	۲۴
Greedy	۶۲/۵۰	۵۹/۱۸	۴۴/۳۳	۳۵/۳۹	۳۷/۹۰	۲۴/۶۰
Q-sch	۵۸/۷۵	۵۹/۱۸	۴۱/۵۸	۳۳/۰۲	۳۷/۳۹	۲۴

در بین الگوریتم‌های رقیب، Random و Mix عملکرد خوبی نداشته‌اند، با این وجود می‌توان گفت Mix نسبت به Random عملکرد بهتری داشته است. روش‌های FIFO و Greedy عملکرد مناسب‌تری نسبت به دو روش Random و Mix داشته‌اند که این دو به طور میانگین در این سناریو عملکرد یکسانی داشته‌اند و روش Q-sch بهترین عملکرد را نسبت به سایر الگوریتم‌های رقیب داشته است. روش پیشنهادی بهترین بهبود را نسبت به روش Random حدود۷۲ درصد با ۱۰ وظیفه و بدترین بهبود را نسبت به روش FIFO و Q-sch حدود ۲۴ درصد با ۲۰ وظیفه داشته است. در کل میزان بهبود زمان انتظار در روش پیشنهادی نسبت به سایر روش‌ها بسیار قابل توجه است و این نشان‌دهنده عملکرد بسیار خوب روش پیشنهادی است.

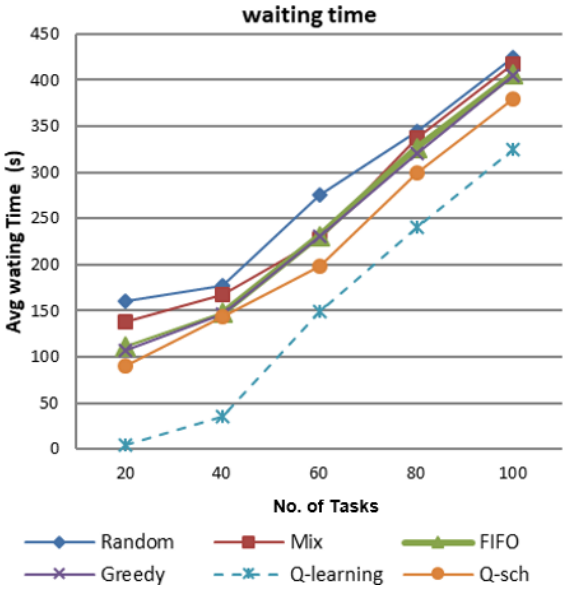
میانگین زمان انتظار وظایف در سناریو دوم در جدول ۱۳ نشان داده شده است. در این سناریو با وجود تعداد وظایف بیشتر، باز هم الگوریتم پیشنهادی بهترین عملکرد را نسبت به سایر روش‌ها داشته است و با افزایش تعداد وظایف، کاهش زمان انتظار را همچنان حفظ کرده است. به طوریکه طبق جدول ۱۳ روش پیشنهادی با تعداد ۲۰ وظیفه، میانگین زمان انتظار را از ۱۶۰ ثانیه به ۴ ثانیه رسانده است، با افزایش تعداد وظایف هم توانسته است، میانگین زمان انتظار را از ۴۲۴ ثانیه به ۳۲۴ ثانیه کاهش دهد. روش Q-sch از آنجایی که هر دو پارامتر میانگین زمان انتظار و ظرفیت بافی مانده بافر را به عنوان پاداش مثبت در نظر می گیرد، ممکن است در برخی موارد این دو با هم در تناقض باشند و منجر به نگاشت مناسب وظایف نگردند، از طرفی چون تنها ظرفیت باقی مانده بافر را به عنوان فضای حالت در نظر می‌گیرد نمی‌تواند درک درستی از محیط به ارسال کننده وظایف بدهد، علت آن هم این است که ارسال کننده وظایف در مرحله‌های قبلی با این حالت با چیدمان متفاوتی از وظایف برخورد کرده است و در اینجا عامل نمی‌تواند تمایزی بین این حالت فعلی و حالتی که قبلاً ملاقات شده قائل شود و در نتیجه منجر به تخصیص نامناسب وظایف بر روی ماشین‌های مجازی می‌گردد. این خطا در تعداد بالایی از وظایف بیشتر اتفاق می‌افتد و منجر به افزایش زمان انتظار وظایف در صف ماشین مجازی و در نتیجه افزایش زمان پاسخ می گردد. روش پیشنهادی با دادن جزئیات بیشتری از محیط به عامل این مشکل را برطرف می‌کند، همچنین به منظور جلوگیری از افزایش فضای حالت از منطق فازی برای گسسته‌سازی استفاده می‌کند.

www.SID.ir

Archive of SID

جدول ۱۳ : میانگین زمان انتظار سناریو دوم (ثانیه)

تعداد وظایف / الگوریتم	۲۰	۴۰	۶۰	۸۰	۱۰۰
Random	۱۶۰	۱۷۷	۲۷۵	۳۴۴	۴۲۴
Mix	۱۳۸	۱۶۶	۲۲۷	۳۳۷	۴۱۷
FIFO	۱۱۱	۱۴۷	۲۳۱	۳۲۶	۴۰۶
Greedy	۱۰۷	۱۴۶	۲۳۱	۳۲۰	۴۰۴
Q-learning	۴	۳۵	۱۴۸	۲۳۹	۳۲۴
Q-sch	۹۰	۱۴۳	۱۹۸	۲۹۹	۳۷۸



شکل ۱۰: میانگین زمان انتظار سناریو دوم

طبق جدول ۱۴ روش پیشنهادی توانسته است تا ۹۰ درصد میانگین زمان انتظار را نسبت به سایر روش‌ها کاهش داده است. پس می‌توان نتیجه گرفت روش پیشنهادی با کاهش حداکثری زمان انتظار وظایف در صف بافر هر ماشین مجازی در تعداد وظایف بالا هم بسیار خوب عمل می‌کند و عملکرد خود را با تعداد وظایف بالا هم حفظ می‌کند.

جدول۱۴: بهبود میانگین زمان انتظار روش پیشنهادی در سناریو دوم (درصد)

تعداد وظایف / الگوریتم	۲۰	۴۰	۶۰	۸۰	۱۰۰
Random	۹۷/۵۰	۸۰/۲۲	۴۶/۱۸	۳۰/۵۲	۲۳/۵۸
Mix	۹۷/۱۰	۷۸/۹۱	۳۴/۸۰	۲۹/۰۸	۲۲/۳۰
FIFO	۹۶/۳۹	۷۶/۱۹	۳۵/۹۳	۲۶/۶۸	۲۰/۱۹
Greedy	۹۶/۲۶	۷۶/۰۲	۳۵/۹۳	۲۵/۳۱	۱۹/۸۰
Q-sch	۹۵/۵۵	۷۵/۵۲	۲۵/۲۵	۲۰/۰۶	۱۴/۲۸

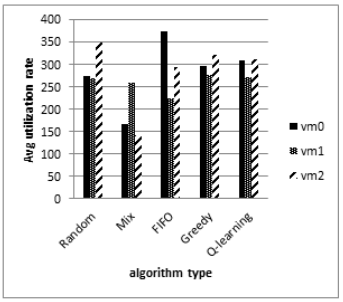
۵-۶- میانگین نرخ بهره‌وری و توزیع بارکاری

در سناریو اول میانگین نرخ بهره‌وری و همچنین میزان توزیع بار کاری بر روی ماشین‌های مجازی موجود از شماره ۰ تا ۲ در سناریو اول برای ۲۰ وظیفه بین روش پیشنهادی و سایر روش‌ها اندازه‌گیری شده است و نتایج با توجه به شکل ۱۱ و شکل ۱۲ به شرح زیر است:

الگوریتم FIFO وظایف را به ترتیب وارد سیستم می‌کند و منجر به توزیع متوازن بارکاری بر روی منابع می‌گردد ولی از آنجا که کارهای کوچک منتظر کارهای طولانی می‌گردند، عادلانه نیست و بهره‌وری از منابع را پایین می‌آورد. الگوریتم Greedy رویکردی مشابه الگوریتم FIFO دارد. نتایج نشان می‌دهد علاوه بر اینکه توزیع عادلانه بارکاری بر روی منابع ایجاد می‌کند، بلکه نرخ بهره‌وری بهتری نسبت به الگوریتم FIFO دارد. الگوریتم‌های mix و Random به دلیل انتخاب تصادفی ماشین‌های مجازی برای اجرای وظایف، نرخ بهره‌وری و توزیع بارکاری مناسبی ندارند. الگوریتم پیشنهادی با اینکه بارکاری را به اندازه الگوریتم FIFO و Greedy به خوبی توزیع نمی‌کند ولی نرخ بهره‌وری بیشتری نسبت به دیگر الگوریتم‌های زمان‌بندی دارد.

شکل ۱۱ میانگین نرخ بهره‌وری و شکل ۱۲ میانگین توزیع بار کاری را روی ماشین‌ها در سناریو اول نشان می‌دهند. همان‌طور که مشخص است الگوریتم FIFO بر روی ماشین مجازی شماره صفر بیشترین بهره‌وری را در حدود ۳۷۳ واحد داشته است سپس الگوریتم پیشنهادی با ۳۰۸ واحد در ردیف بعدی قرار می‌گیرد. همچنین بیشترین بهره‌وری روی ماشین شماره یک متعلق به الگوریتم Greedy و سپس الگوریتم پیشنهادی به ترتیب ۲۷۴ و ۲۷۰ است. بر روی ماشین شماره دو هم بیشترین بهره‌وری به ترتیب متعلق به الگوریتم‌های Random و روش پیشنهادی با مقدارهای ۳۴۷ و ۳۱۸ است. روش پیشنهادی در میان روش‌ها بر روی هیچ کدام از ماشین‌ها بیشترین نرخ بهره‌وری را نداشته است ولی به طور میانگین نرخ بهره‌وری روش پیشنهادی برابر ۲۹۹/۲۴۴، روش Greedy برابر ۲۹۳/۹۰ ، روش FIFo برابر ۱۷۱/۱۹۸، روش Mix برابر ۱۸۹/۱ و روش Random برابر ۲۹۶/۵۹ است. پس می‌توان نتیجه گرفت الگوریتم پیشنهادی به عنوان هدف اصلی نه تنها میانگین زمان پاسخ، زمان انتظار و زمان تکمیل کار را بهبود می‌بخشد بلکه نرخ بهره‌وری بیشتری نیز نسبت به سایر روش‌های زمان‌بندی موجود دارد.

شماره vm / الگوریتم	0	1	2
Random	274	268	348
Mix	168	259	141
FIFO	374	223	293
Greedy	296	275	311
Q-learning	309	271	318

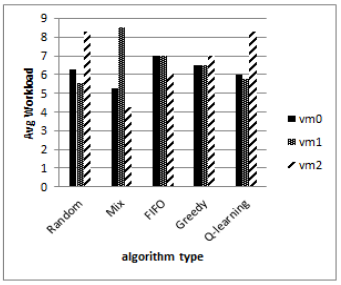


شکل ۱۱ : میانگین نرخ بهره‌وری در سناریو اول

www.SID.ir

Archive of SID

شماره vm / الگوریتم	0	1	2
Random	6.25	5.5	8.25
Mix	5.25	8.5	4.25
FIFO	7	7	6
Greedy	6.5	6.5	7
Q-learning	6	5.75	8.25



شکل۱۲: میانگین توزیع بارکاری در سناریو اول

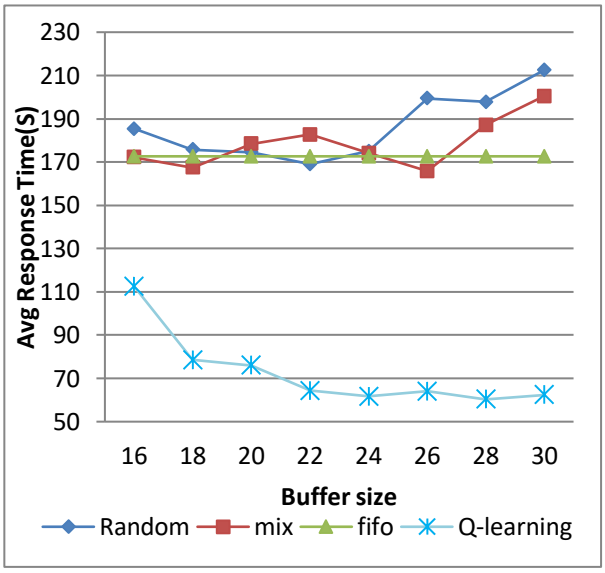
در سناریو اول میانگین نرخ بهره‌وری را بر روی هر ماشین مجازی به صورت جداگانه در روش پیشنهادی و سایر روش‌های زمان‌بندی محاسبه کردیم. در سناریو دوم میانگین نرخ بهره‌وری را نسبت به کل ماشین‌های مجازی برای ۲۰، ۴۰، ۶۰، ۸۰ و ۱۰۰ وظیفه تحت سناریو دوم، در روش پیشنهادی و دیگر روش‌ها مورد تحلیل قرار می‌دهیم. همان‌طور که در شکل ۱۳ قابل مشاهده است با افزایش تعداد وظایف کاربران باز هم الگوریتم پیشنهادی نسبت به سایر روش‌ها بیشترین میانگین نرخ بهره‌وری را دارا است. البته این نکته حائز اهمیت است که وظایف موجود در این سناریو بسیار کوچک هستند و این اختلاف ناچیز در میزان نرخ بهره‌وری میان الگوریتم پیشنهادی و سایر الگوریتم‌ها بسیار با اهمیت است.

۵-۷- تأثیر افزایش ظرفیت بافر بر میانگین زمان پاسخ

در این بخش الگوریتم پیشنهادی با سه الگوریتم Random، Mix و FIFO تحت ۴۰ وظیفه با ۳ ماشین مجازی با ظرفیت‌های بافر متفاوت مورد مقایسه قرار گرفته است. همان‌طور که شکل ۱۴ نشان می‌دهد. میانگین زمان پاسخ با توجه به افزایش ظرفیت بافر ماشین‌های مجازی در الگوریتم Random و Mix افزایش می‌یابد. الگوریتم FIFO نسبت به افزایش ظرفیت بافر حساس نیست و میانگین زمان پاسخ آن در تمام مراحل یکسان است. الگوریتم پیشنهادی در ابتدا نسبت به افزایش ظرفیت بافر حساس است ولی با افزایش ظرفیت بافر، میزان حساسیت آن کم می‌شود و از ظرفیت خاصی به بعد هیچ تأثیری در میانگین زمان پاسخ ندارد. این الگوریتم با افزایش ظرفیت بافر برخلاف الگوریتم‌های دیگر منجر به کاهش زمان پاسخ می‌گردد. روش پیشنهادی حتی در مراحلی که نسبت به افزایش ظرفیت بافر حساس است، مثل ۱۶، ۱۸ و ۲۰ باز هم میانگین زمان پاسخ آن نسبت به سایر روش‌های موجود بسیار کمتر است بطوریکه طبق جدول ۱۵ هنگامی که ظرفیت بافر ۱۶ است، میانگین زمان پاسخ روش پیشنهادی ۱۱۲ ثانیه است، در حالیکه میانگین زمان پاسخ الگوریتم FIFO برابر ۱۸۳ ثانیه، الگوریتم Mix برابر ۱۷۲ ثانیه و الگوریتم Random برابر ۱۸۵ثانیه است.

جدول۱۵: میانگین زمان پاسخ با توجه به افزایش ظرفیت بافر(ثانیه)

بافر / الگوریتم	۱۶	۱۸	۲۰	۲۲	۲۴	۲۶	۲۸	۳۰
Random	۱۸۵	۱۷۶	۱۷۴	۱۶۹	۱۷۵	۱۹۹	۱۹۸	۲۱۲
Mix	۱۷۲	۱۶۷	۱۷۸	۱۸۳	۱۷۴	۱۶۶	۱۸۷	۲۰۰
FIFO	۱۷۳	۱۷۳	۱۷۳	۱۷۳	۱۷۳	۱۷۳	۱۷۳	۱۷۳
Q-learning	۱۱۲	۷۸	۷۶	۶۴	۶۲	۶۴	۶۰	۶۲



شکل ۱۴: میانگین زمان پاسخ با توجه به افزایش ظرفیت بافر

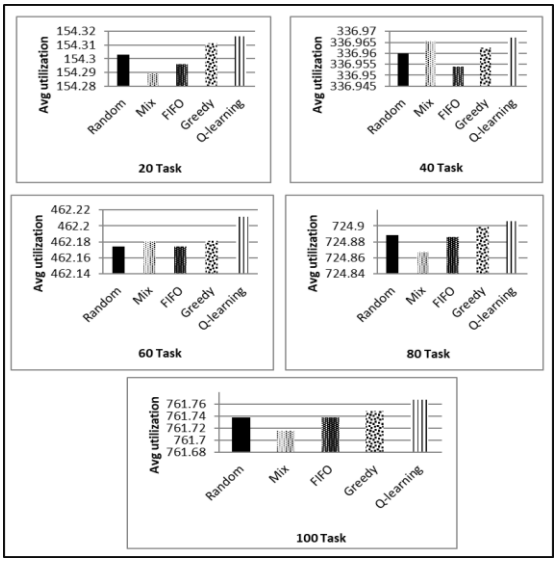
۶- ارزیابی کلی و نتیجه‌گیری

از آنجا محیط ابر یک محیط پویا است و درخواست‌های کاربران در طول زمان متغیر هستند، نیازمند روش زمان‌بندی بهینه‌ای هستیم که بتواند در زمان کمتر درخواست‌های بیشتری را پردازش کند. یادگیری تقویتی قابلیت تطبیق با چنین محیطی را دارد و می‌تواند در اثر یادگیری، بهترین نگاشت وظایف به منابع را انجام دهد و وظایف را به صورت کارا زمان‌بندی کند از این رو یک روش زمان‌بندی جدید مبتنی بر روش یادگیری کیو که یکی از پرکاربردترین روش‌های یادگیری تقویتی است، ارائه شد. این روش پیشنهادی با پنج روش زمان‌بندی دیگر (Random-Mix-FIFO-Greedy-Q-sch) در محیط شبیه ساز کلودسیم مورد ارزیابی و مقایسه قرار گرفت، نتایج حاصل از آزمایش‌ها در بازه‌های مختلف بیانگر عملکرد بهتر و افزایش کیفیت سرویس روش پیشنهادی نسبت به سایر روش‌ها است که باعث به حداقل رساندن زمان انتظار، زمان پاسخ، زمان تکمیل کار و همچنین افزایش نرخ بهره‌وری می‌گردد. نتایج حاصل از شبیه‌سازی نشان می‌دهند که روش پیشنهادی توانسته است به طور میانگین در محدوده تعداد وظایف بالا، میانگین زمان تکمیل کار را نسبت به روش Random حدود ۲۷/۹۳ درصد، نسبت به روش Mix ۲۲/۶۰ درصد، نسبت به روش FIFO ۲۰/۷۰ درصد، نسبت به روش Greedy ۱۷/۵۵ درصد و نسبت به روش Q-sch ۱۲/۷۶ درصد بهبود

www.SID.ir

Archive of SID

ببخشد. روش پیشنهادی بهبود بیشتری در میانگین زمان پاسخ نسبت به میانگین زمان تکمیل کار نسبت به سایر روش‌ها داشته است به طوری که نسبت به روش Random ۴۹/۵۲ درصد، نسبت به Mix ۴۶/۰۳ درصد، نسبت به FIFO ۴۳/۹۹ درصد و نسبت به به روش‌های Greedy و Q-sch به ترتیب توانسته است به ترتیب ۴۳/۵۳ و ۳۸/۶۸ درصد بهبود به دست بیاورد. در میانگین زمان انتظار هم با کاهش حداکثری زمان انتظار در صف بافر ماشین‌های مجازی نیز بیشترین بهبود را نسبت به دو پارامتر میانگین زمان تکمیل کار و زمان پاسخ داشته است که نتایج نشان می دهند روش پیشنهادی توانسته است نسبت به روش Random ۵۵/۶ درصد، نسبت به روش Mix ۵۲/۴۳ درصد، نسبت به روش FIFO ۵۱/۰۷ درصد، نسبت به روش Greedy ۵۰/۶۶ درصد و نسبت به روش Q-sch ۴۶/۱۳ درصد بهبود حاصل کند. همچنین در هر مرحله نرخ بهره‌وری بیشتری نسبت به دیگر روش‌های زمان‌بندی ذکر شده به دست آورد. با وجود استفاده فراوان از روش یادگیری کیو در مسائل، این روش معایبی مثل افزایش فضای حالت و مشکلات ذخیره‌سازی دارد بطوریکه با افزایش فضای حالت، برخی حالت‌ها توسط عامل مشاهده نمی‌شوند و همچنین باعث می‌شود، روش زمان‌بندی از بار پردازشی بالا رنج ببرد که ممکن است تأثیر منفی بر روی سرعت همگرایی، عدم انتخاب مناسب اعمال و ارزش‌های بهینه‌سازی داشته باشد. با اینکه روش پیشنهادی از منطق فازی برای کاهش فضای حالت را استفاده می‌کند، می‌توان از روش‌های دیگری هم برای سرعت بخشیدن بیشتر در یادگیری استفاده کرد. همچنین در این مقاله منابع همگن در نظر گرفته شده‌اند که در آینده قصد داریم این روش را با وجود منابع ناهمگن نیز بررسی کرده، همچنین هزینه‌ها را به عنوان دیگر پارامتر کیفیت سرویس مورد ارزیابی قرار دهیم.



شکل۱۳: میانگین نرخ بهره‌وری در سناریوی دوم با تعداد وظایف ۲۰ تا ۱۰۰

مراجع

[1] A. A. Buhussain, R. E. D. Grande and A. Boukerche, "Performance analysis of Bio-Inspired scheduling algorithms for cloud," IEEE International parallel and distributed processing symposium workshops, pp. 776-785, 2016.

[2] Z. Peng, D. Cui, J. Zuo, Q. Li, B. Xu and W. Lin, "Random task scheduling scheme based on reinforcement learning in cloud computing," Cluster computing, vol. 18, pp. 1595-1607, 2015.

[3] E. Barrett, E. Howley and J. Duggan, "Applying reinforcement learning towards automating resource allocation and application scalability in the cloud," Concurrency and computation: practice and experience, vol. 25, no. 12, pp. 1656-1674, 2013.

[4] B. Yang, X. Xu, F. Tan and D. H. Park, "An utility-based job scheduling algorithm for cloud computing considering reliability factor," Proceeding of international conference on cloud and service computing, pp. 95-102, 2011.

[5] M. Hussin, C. Young and A. Zomaya, "Efficient energy management using adaptive reinforcement learning-based scheduling in large-scale distributed systems," Proceeding of the international conference on parallel processing, pp. 385-393, 2011.

[6] M. Hussin, N. Hamid and K. Kasmiran, "Improving reliability in resource management through adaptive reinforcement learning for distributed systems," Journal of parallel and distributed computing, vol. 75, pp. 93-100, 2015.

[7] Z. Peng, D. Cui, Y. Ma, J. Xiong, B. Xu and W. Lin, "A Reinforcement Learning-based Mixed Job Scheduler Scheme for Cloud," in IEEE transactions on cloud computing, 2017.

[8] M. Cheng, J. Li and S. Nazarian, "DRL-Cloud: Deep Reinforcement Learning-Based Resource," in Proceedings of the 23rd Asia and south pacific design automation conference, pp. 129/134, 2018.

[9] Y. Zhao, X. Mingqing and G. Yawei, "Dynamic resource scheduling of cloud-based automatic test system using reinforcement learning" in IEEE 13th international conference on electronic measurement & instruments, pp. 159-165, 2017.

[۱۰] سیمین قاسمی فلاورجانی و محمد علی نعمت بخش و بهروز شاهقلی قهفرخی، « تخصیص وظایف چندهدفه در واگذاری به ابر سیار،» مجله مهندسی برق دانشگاه تبریز، دوره ۴۶، شماره ۴، صفحات ۲۱۷-۲۳۲، ۱۳۹۵.

[۱۱] شهرام جمالی و سپیده ملک تاجی و مرتضی آنالویی، « مکان‌یابی ماشین‌های مجازی با استفاده از الگوریتم رقابت استعماری،» مجله مهندسی برق دانشگاه تبریز، دوره ۴۶، شماره ۱، صفحات ۵۳-۶۲، ۱۳۹۵.

[12] R. S. Sutton and A. G. Barto, *Reinforcement Learning: an introduction*, Cambridge press, 1998.

[13] L. P. Kaelbling, M. L. Littman and A. W. Moore, "Reinforcement learning: a survey," Artificial intelligence research, no. 4, pp. 237-285, 1996.

www.SID.ir