Karafan

Quarterly Scientific Journal of Technical and Vocational University

Autumn 2022, Vol. 19, No. 3, p. 529-551

Journal Homepage: https://karafan.tvu.ac.ir/?lang=en

TVU

doi:10.48301/KSSA.2022.341879.2104

A Prediction based Proactive Resource Provisioning Strategy for Multi-objective Workflow Scheduling in Cloud Computing

Reza Mahmoudian[1], Reihaneh Khorsand[2*], Mohammadreza Ramezanpour[3]

[1]M.Sc. Student, Department of Computer Engineering, Dolatabad Branch, Islamic Azad University, Isfahan, Iran.

[2]Associate Professor, Department of Computer Engineering, Dolatabad Branch, Islamic Azad University, Isfahan, Iran.

[3]Associate Professor, Department of Computer Engineering, Mobarakeh Branch, Islamic Azad University, Isfahan, Iran.

ARTICLE INFO

Article Type:
Original Research

Received: 05.23.2022
Revised: 11.04.2022
Accepted: 11.19.2022

Keyword:
Cloud Computing
Multi-objective Scheduling
Scalability
Quality of Service
LVQ

***Corresponding Author:**
Reihaneh Khorsand
Email:
reihaneh_khm@yahoo.com

ABSTRACT

In order to manage the workload proactively on the Cloud system during application execution, workload should be predicted through a proper approach and count of resources handled through an auto-scaling system controller. On the other hand, the workflow scheduling requires proper mapping of cloud resources to workflow tasks to efficiently utilize resources and meet different user's quality of service requirements. Workflow scheduling is NP-complete problem and multi-objective evolutionary algorithms have shown their merit for solving such problem. Most of the works in the literature focused either on dynamic resource provisioning or scheduling algorithms for executing workloads. Based on this deficiency, in this paper, a prediction-based proactive resource provisioning strategy based on learning vector quantization (LVQ) artificial neural network was proposed to predict the workloads in future and a fuzzy system controller was proposed to compute the proper number of resources to be allocated to the Cloud system. In addition, the multi-objective linear programming scheduling algorithm was proposed to execute workloads effectively on available resources. An evaluation with three kinds of real scientific workflows was performed. The experimental results showed that the proposed approach efficiently reduced execution average cost, and response time along with higher resource utilization in comparison with its counterparts.

©2022 the authors. Published by Technical and Vocational University, Tehran, Iran. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution-Noncommercial 4.0 International (CC BY-NC License) (https://creativecommons.org/licenses/by-nc/4.0/)

E-ISSN: 2538-4430
ISSN: 2382-9796

EXTENDED ABSTRACT

Introduction

In cloud computing, task scheduling with service level agreements while resource provisioning is considered dynamically a challenging issue. Task scheduling is a basic method in cloud computing that is responsible for the distribution of computing tasks among the pool of virtual resources and it is an important issue. A great deal of research in this field has been carried out but most of proposed methods is considered an objective for optimization. In general, the cloud service providers have different objectives, where they might have conflicting natures and must be met simultaneously. On the other hand, one of the most important challenges of task scheduling in the cloud computing is resource scalability, which allows timely resources provisioning to meet application requirements. Therefore, cloud service providers need approaches to perform automatic scaling in the cloud by considering all effective factors.

Most of the existing methods for optimal task scheduling among different virtual machines of cloud data centers are not acceptable to achieve reasonable service quality. In fact, it is difficult to consider all the objectives of the desired service level agreements along with providing a dynamic resource for the implementation of tasks. In this paper, an active resource provisioning method based on prediction is introduced for multi-objective workflow scheduling in cloud computing, which uses automatic resource provisioning in cloud computing. The framework of the proposed method was based on two important parts of dynamic resource provisioning and task scheduling. The objectives were to improve the response time, cost and average utilization of the virtual machine in comparison with other methods.

Methodology

In the proposed framework, the user's request is in the form of a workflow, which includes a set of partially interdependent tasks. The cloud broker has two important parts: dynamic resource provisioning and workflow task scheduling, which are performed autonomously. In general, the request information and the amount of current workload are collected by the monitoring phase. Then, the amount of future workload is predicted using LVQ neural network, and the amount of SLA violation is also calculated and placed in the knowledge base. The decision-making phase by using knowledge base, including the degree of SLA violation and predicted workload, applies a fuzzy algorithm to decide how many new virtual machines to add or remove from the system. This is called horizontal scaling.

Results and discussion

To simulate the proposed method in the cloud environment, CloudSim 3.0.3 tool was used. Three well-known scientific workflow structures named Montage, Cybershake and Epigenomics were used. Utilization, average response time and total cost were considered to evaluate the performance of the proposed method. In addition, the proposed method was compared with PAPS and SAS algorithms. According to the evaluated results, using linear programming and creating cost constraints for proper resource allocation and also using

the objective function to minimize the cost provided greater priority to the cost of the proposed method in resource allocation so that the proposed method performed better than the compared methods in all different workflows. However, the most important reason in reducing the cost of using a suitable fuzzy system for resource provision based on exceeding the service conditions was the cost, which increased the accuracy in adding and subtracting virtual machines.

Due to the existence of a fuzzy decision maker and the use of two important input variables, which are predicted response time and workload respectively, the resource provisioning parameter in the proposed method showed superiority than the other methods. If the appropriate resource was not provided, there would be a pending request that will remain without a resource, and the response time would increase. Moreover, the optimal performance of linear programming as a recognition of the prerequisites and decision-making conditions of the system in the resource provisioning phase led to a more accurate selection of the resource and reduced the response time compared to other methods.

The simulation results showed that the proposed method performed better in terms of utilization compared to the other two methods. Careful monitoring of resources in the cloud and continuous diagnosis of workload was a very effective factor for controlling the status of resources in the proposed method. In fact, due to the monitoring of the CIS status and the use of fuzzy decision-making to provide resources based on the user's requests, scaling was carried out optimally.

Conclusion

In the present research, a task scheduling method based on autonomous resources provisioning was proposed, performing auto scaling in the cloud by considering all effective factors. In the proposed method, user requests are converted into partial tasks in the form of a workflow. The tasks are assigned to the CI where has two important parts to provide dynamic resource provisioning and task scheduling. After deciding on the resource provisioning, the change commands are submitted to the CIS unit so that this unit can notify the CI of the changes to be carried out. After receiving the information from CIS, the task scheduling unit allocates the resources after prioritizing the workflow tasks. The proposed method was tested and evaluated on three different workflow structures. The simulation results showed that the proposed method increased utilization by 6.5% on average and reduced response time and cost by 8.3% and 7.9%, respectively.

کارافن
فصلنامه علمی دانشگاه فنی و حرفه‌ای
پاییز ۱۴۰۱، دوره ۱۹، شماره ۳، ۵۵۱-۵۲۹
آدرس نشریه: https://karafan.tvu.ac.ir/
doi:10.48301/KSSA.2022.341879.2104

ارائه یک روش تأمین منبع فعالانه مبتنی‌بر پیشگویی برای زمان‌بندی گردش کار چندهدفه در رایانش ابری

رضا محمودیان[۱]، ریحانه خورسند[۲*]، محمدرضا رمضان‌پور[۳]

۱- دانشجوی کارشناسی ارشد، گروه مهندسی کامپیوتر، واحد دولت آباد، دانشگاه آزاد اسلامی، اصفهان، ایران.
۲- دانشیار، گروه مهندسی کامپیوتر، واحد دولت آباد، دانشگاه آزاد اسلامی، اصفهان، ایران.
۳- دانشیار، گروه مهندسی کامپیوتر، واحد مبارکه، دانشگاه آزاد اسلامی، اصفهان، ایران.

اطلاعات مقاله

نوع مقاله: مقاله پژوهشی

دریافت مقاله: ۱۴۰۱/۰۳/۰۲
بازنگری مقاله: ۱۴۰۱/۰۸/۱۳
پذیرش مقاله: ۱۴۰۱/۰۸/۲۸

کلید واژگان:
رایانش ابری
زمان‌بندی چندهدفه
خاصیت مقیاس‌پذیری
کیفیت سرویس
LVQ

***نویسنده مسئول:** ریحانه خورسند
پست الکترونیکی:
reihaneh_khm@yahoo.com

چکیده

به‌منظور مدیریت فعالانه بار کاری در طول اجرای برنامه کاربردی روی سیستم ابر، بار کاری باید از طریق یک روش مناسب پیشگویی شود و از طریق یک کنترل‌کننده سیستم مقیاس‌پذیر تعداد منابع موردنیاز مدیریت شوند. از سوی دیگر زمان‌بندی گردش‌های کار به‌منظور بهره‌وری منابع و برآورده کردن نیازمندی‌های کیفی کاربران مختلف به نگاشت مناسب منابع ابری به وظایف گردش کار نیاز دارد. زمان‌بندی گردش کار یک مسئله NP کامل است و الگوریتم‌های تکاملی چندهدفه اغلب برای حل این مسائل مفید هستند. بیشتر کارهای گذشته تنها بر تأمین منبع یا زمان‌بندی اجرای گردش کارها تمرکز کرده‌اند و تاکنون به ترکیب پویای آن‌ها توجه نشده است. براساس کمبودی که در این زمینه وجود دارد، در این مقاله یک استراتژی تأمین منبع فعالانه با استفاده از شبکه عصبی رقمی‌ساز بردار یادگیر (LVQ) برای پیشگویی بارهای کاری آینده ارائه می‌شود و یک کنترل‌کننده سیستم فازی برای محاسبه تعداد مناسب منابع موردنیاز پیشنهاد می‌شود که این منابع باید به سیستم اختصاص داده شوند. همچنین یک الگوریتم زمان‌بندی مبتنی بر روش برنامه‌ریزی خطی چندهدفه برای اجرای بارهای کاری روی منابع موجود پیشنهاد می‌شود. در نهایت ارزیابی مقاله با سه نوع بارکاری واقعی انجام می‌شود. نتایج آزمایش‌ها نشان می‌دهند که روش پیشنهادی متوسط هزینه اجرا و زمان پاسخ را در مقایسه با کارهای انجام شده دیگر، کاهش و میزان بهره‌وری منابع را افزایش می‌دهد.

شاپای الکترونیکی: ۲۵۳۸-۴۴۳۰
شاپای چاپی: ۲۳۸۲-۹۷۹۶

©2022 the authors. Published by Technical and Vocational University, Tehran, Iran. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution-Noncommercial 4.0 International (CC BY-NC License) (https://creativecommons.org/licenses/by-nc/4.0/)

مقدمه

رایانش ابری، فناوری نوظهوری است که بیشتر سازمان‌ها به‌نحوی سعی دارند از آن در استراتژی‌های کسب‌وکار خود استفاده کنند. در حال حاضر بسیاری از شرکت‌های سنتی در حال تغییر ساختار خود به ساختاری می‌باشند که در آن بتوانند از قابلیت فناوری‌هایی با زیرساخت رایانش ابری استفاده کنند [۱]. در حقیقت، رایانش ابری براساس منابع با قابلیت انعطاف‌پذیری، مقیاس‌پذیری و دسترس‌پذیری بالا و هزینه تعمیر و نگهداری کم و ... فعالیت می‌کند [۲].

در حوزه رایانش ابری، عملیات زمان‌بندی وظایف با آگاهی از توافقات در قبال خدمات به‌طوری‌که تأمین منابع به‌صورت پویا در نظر گرفته شود یک موضوع بحث‌برانگیز است. زمان‌بندی وظیفه، یک روش پایه‌ای در محاسبات ابری است که توزیع وظایف محاسباتی در میان استخر منابع مجازی را به عهده دارد و موضوع مهمی است که تاکنون تحقیقات زیادی در این زمینه انجام شده است ولی هدف بیشتر آن روش‌ها بهینه‌سازی یک هدف بوده است. به‌طور کلی فراهم‌کننده سرویس ابر و سرویس‌گیرنده‌ها اهداف و نیازمندی‌های متفاوتی دارند که این نیازمندی‌های چندگانه ممکن است ماهیت متضاد داشته باشند و باید به‌طور هم‌زمان برآورده شوند. برای نمونه، حداقل کردن زمان اتمام کار و نقض مهلت زمانی روی تعداد زیادی وظایف دشوار است اگر که خواهان کاهش هزینه‌ها نیز باشیم.

از سوی دیگر یکی از مهم‌ترین معضلات زمان‌بندی وظیفه در محیط ابر، مقیاس‌پذیری منابع است که تأمین به‌هنگام منابع را برای ملاقات نیازمندی‌های برنامه کاربردی اجازه می‌دهد [۳]. بدین صورت که به‌جای رزرو قبلی منابع موردنیاز، زمان‌بندی وظیفه را مجبور به استفاده از خاصیت کشسانی منابع ابری می‌کند. درواقع در بستر رایانش ابری میلیون‌ها کاربر با یکدیگر یا با سرویس‌دهنده‌های شرکت‌های فراهم‌کننده اینترنتی در ارتباط هستند [۴]. کاربران برای داشتن یک ارتباط رضایت‌مند به خدمات کارآمد نیاز دارند که این خدمات را شرکت‌های فراهم‌کننده سرویس آماده و به کاربران ارائه می‌کنند [۵]. با توجه به اینکه استفاده از منابع ابری، براساس توافق هزینه در قبال سرویس می‌باشد ضعف و ناکارآمدی روش‌های تأمین منبع، کاربران ابری را با مسئله افزایش هزینه استفاده و ارائه‌دهندگان را با مشکل نارضایتی مشتریان مواجه خواهد کرد. بنابراین ارائه‌دهندگان سرویس‌های ابری، نیازمند رویکردهایی هستند که با در نظر گرفتن تمامی عوامل مؤثر، مقیاس‌بندی خودکار در ابر را انجام دهند.

بیشتر روش‌هایی که برای زمان‌بندی بهینه وظایف گردش کار در بین ماشین‌های مجازی مختلف مراکز داده ابر موجود هستند برای حصول کیفیت سرویس معقول قابل قبول نیستند [۶; ۷]. درواقع درنظر گرفتن تمام اهداف توافقات سطح خدمات موردنظر به همراه تأمین منبع پویا برای اجرای وظایف کار دشواری می‌باشد. برای حل مشکلات مطرح‌شده، در این مقاله یک استراتژی تأمین منبع فعالانه مبتنی بر پیشگویی برای زمان‌بندی گردش کار چندهدفه در رایانش ابری معرفی می‌شود که از تأمین منبع خودکار در محیط ابر استفاده می‌کند. چارچوب استراتژی پیشنهادی بر پایه دو بخش مهم تأمین پویای منابع و زمان‌بندی گردش کار پایه‌گذاری شده است. هدف بهبود زمان پاسخ، هزینه و میانگین بهره‌وری ماشین مجازی در مقایسه با روش‌های دیگر است.

ادامه مقاله به‌صورت زیر سازماندهی شده است: بخش ۲ مرور کلی تحقیقات مرتبط می‌باشد. در بخش ۳ روش پیشنهادی که شامل تأمین پویای منابع و زمانبندی گردش کارها می‌باشد با جزئیات کامل بیان می‌شود. بخش ۴ ارزیابی عملکرد روش پیشنهادی را فراهم می‌کند. نتیجه‌گیری و پژوهش‌های آینده در بخش ۵ مورد بحث قرار می‌گیرند.

پیشینه تحقیق

در دهه گذشته چندین چارچوب، تکنیک، روش، مدل و ... برای زمان‌بندی گردش کارها و تأمین منبع پویا و ایستا ارائه شده است [۸-۱۲]. در تأمین ایستای منابع برای زمان‌بندی گردش کار، میزان ظرفیت تأمین‌کننده در طول زمان، تغییری نمی‌کند و معمولاً حداکثر منابع موردنیاز (به اندازه زمان اوج تقاضا) برای زمان‌بندی گردش کارها در تمام اوقات تأمین می‌شود [۱۳]. در این نوع روش تأمین منبع، در بیشتر اوقات منابع به‌هدر می‌روند زیرا میزان بار کاری در تمام

اوقات به اندازه زمان اوج تقاضا نیست اما تأمین‌کننده باید حداکثر منابع موردنیاز را برای جلوگیری از تخطی توافقات سطح خدمات از قبل تأمین کرده باشد. این مدل الگوریتم‌ها نیازمند اطلاعات اضافی در مورد تعداد درخواست‌ها و اطلاعاتی در مورد منابع در حال اجرا نیز می‌باشند [۱۴; ۱۵]. در اینجا نیازی به نظارت مستمر منابع نیست زیرا این مدل‌ها فقط زمانی‌که بار کاری با تغییرات اندکی مواجه است به‌خوبی کار می‌کنند.

در پژوهش خورسند و همکاران [۱۶] یک استراتژی به نام ATSDS معرفی می‌شود که زمان‌بندی گردش کارها را در دو مرحله قطعه‌بندی گردش کار و زمان‌بندی قطعه‌ها انجام می‌دهد. در مرحله اول، گردش‌های کار با در نظرگرفتن شرایط زمان اجرای ابر یعنی براساس تعداد ماشین‌های مجازی و متوسط پهنای باند موجود بین آنها به‌طور پویا قطعه‌بندی می‌شوند. سپس در مرحله دوم، قطعه‌های گردش کار ایجادشده، طبق مهلت زمانی گردش کار و ظرفیت ماشین‌های مجازی، برای اجرا به ماشین‌های مجازی تخصیص داده می‌شوند. در تحقیق آنها گردش‌های کار مبتنی بر نمونه شامل محدودیت مهلت زمانی به‌صورت بی‌درنگ زمان‌بندی می‌شوند اما از یک تأمین منبع ایستا برای تخصیص وظایف گردش کار استفاده می‌کند و انعطاف‌پذیری کمی را فراهم می‌کند.

دومین روشی که معمولاً در مراکز داده ابری به‌کار گرفته می‌شود و البته بیشتر مورد توجه است تأمین منابع به‌صورت پویا برای تضمین خاصیت کشسانی در رایانش ابری است. در تأمین پویای منابع (تأمین مبتنی بر تقاضا)، ظرفیت تأمین‌کننده در طول زمان متناسب با میزان تقاضای کاربران، کم و زیاد می‌شود. در حقیقت، تأمین پویای منابع، امکان استفاده از مدل پرداخت- به میزان استفاده را برای ارائه‌دهندگان ابری فراهم می‌کند. برخلاف مدل ایستا، در مدل پویا دائماً منابع ابری نظارت می‌شوند تا در برابر تغییرات بار کاری به‌خوبی عمل کنند. با همه مزایایی که تأمین پویای منابع دارد نباید فراموش کرد که استقرار و اجرای آن، معضلات و مشکلاتی را به دنبال خواهد داشت. برای مثال برای برنامه‌ریزی تأمین منابع باید زمان‌های مناسب تأمین منابع، درنظر گرفته شود. اگر منابع، خیلی زودتر از موعد مناسب تأمین شوند باعث اتلاف منابع و در نتیجه، افزایش هزینه خواهند شد. از طرفی دیگر، اگر منابع، دیرتر از موعد (خیلی دیر) تأمین شوند به‌طور بالقوه، تخطی‌های توافقات سطح سرویس را در پی خواهند داشت و باعث نارضایتی کاربران خواهند شد.

اعلایی و همکاران [۱۷] چارچوبی برای تأمین منبع برای خدمات محیط ابری ارائه می‌دهد که از یک تأمین منبع پویا و خودمختار بر پایه حلقه کنترل MAPE استفاده می‌کند. حلقه کنترل MAPE شامل چهار مرحله نظارت، تحلیل، برنامه‌ریزی و اجرا می‌باشد. آنها رویکرد برای تأمین منبع ترکیبی برای خدمات ابری ارائه کرده‌اند که برمبنای ترکیبی از مفاهیم محاسبات خودمختار و یادگیری ماشین می‌باشد. در کار آنها روش تصمیم‌گیری در راستای مقیاس‌بندی برمبنای یادگیری تقویتی می‌باشد که به‌وسیله فرایند تصمیم مارکوف مدل شده است. همچنین استراتژی تخصیص منبع براساس سیاست‌های تعادل بار ساده مثل نوبتی چرخشی، تصادفی و غیره درنظر گرفته شده است که مهلت‌های زمانی را در تخصیص منبع درنظر نمی‌گیرند.

پاک‌نژاد و همکاران [۱۸] روشی برای مقیاس‌پذیری خودکار برنامه‌های کاربردی وب در محیط ابر ارائه کرده‌اند که برمبنای حلقه کنترل MAPE و با رویکرد آگاه از هزینه می‌باشد. تمرکز روش پیشنهادی آنها بر فاز آخر کنترل MAPE یعنی اجراکننده می‌باشد و یک اجراکننده کاهنده هزینه ارائه می‌دهند. برخلاف سایر اجراکننده‌ها، روش ارائه‌شده فرمان‌های کاهش مقیاس را با آگاهی از انتخاب ماشین‌های مجازی مازاد محاسبه می‌کند. علاوه بر آن ماشین‌های مجازی مازاد انتخاب‌شده به‌صورت قرنطینه نگهداری می‌شوند تا در صورت نیاز به افزایش مقیاس در همان دوره ساعتی که صورت‌حساب آن ماشین‌های مجازی پرداخت‌شده آنها را آزاد و در هزینه صرفه‌جویی کند.

بحرپیما و همکاران [۱۹] یک رویکرد کنترل انطباقی برای تأمین منبع پویا ارائه داده‌اند که مقیاس‌پذیری آن برمبنای یادگیری تقویتی مستمر و فراهم آوردن منابع پویا در محیط‌های متغیر ابر است که بدون قطعیت هستند. تأمین‌کننده منبع پویای ارائه‌شده یک کنترل‌کننده هدف است که توانایی مدیریت عدم‌قطعیت را به‌طور خاص دارد. کنترل‌کننده ارائه‌شده به‌عنوان هدف اولیه از رد شدن کار جلوگیری می‌کند و در هدف دوم خود مصرف انرژی را به‌حداقل می‌رساند.

جمشیدی و همکاران [۲۰] یک رویکرد کنترل حد آستانه در راستای تأمین منبع پویا با استفاده از سیستم منطق فازی نوع ۲ ارائه کرده‌اند که یک تأمین منبع خودمختار برای نرم‌افزارهای مبتنی بر ابر فراهم می‌آورد. درواقع در پژوهش آنها مسئله تخصیص پویای منابع برای برنامه‌های مبتنی بر ابر که با کارهای غیرقابل پیش‌بینی روبه‌رو می‌شوند بررسی می‌گردد تا هزینه‌ها را بدون نقض توافقات سطح خدمات کاهش دهد. درواقع در پژوهش آنها یک کنترل‌کننده کشسانی ترکیبی برای تنظیم منابع موردنیاز هنگامی که برنامه کاربردی در حال اجراست ارائه شده است. روش تخصیص منبع در رویکرد آن‌ها براساس هزینه ماشین‌های مجازی می‌باشد که متناسب با زمان دقیق بین دستیابی ماشین تا زمان رهاسازی آن است.

شی و همکاران [۲۱] یک روش چهار مرحله‌ای دارای زمان‌بندی گردش کارهای علمی و تأمین منبع پویا ارائه کردند که محدودیت‌های بودجه‌ای و مهلت زمانی را درنظر می‌گیرد. مراحل کار ارائه‌شده توسط آنها شامل پیش‌پردازش کار، کنترل پذیرش کار، تأمین منبع کشسانی و زمان‌بندی وظایف می‌باشد. هدف از روش ارائه‌شده تکمیل بسیاری از گردش کارهای با اولویت بالا است که ممکن است تحت محدودیت بودجه و مهلت زمانی قرار گرفته باشند.

در مرجع [۲۲] مانند روش ارائه‌شده در [۲۱] نیز با توجه به محدودیت بودجه و مهلت زمانی روشی برای زمان‌بندی گردش کارهای علمی و تأمین منابع به‌صورت پویا ارائه داده‌اند با این تفاوت که از ترکیب الگوریتم سریع‌ترین زمان اتمام و الگوریتم کلونی مورچه برای پیش‌بینی بار کاری و تصمیم‌گیری تعداد منابع به‌صورت پویا استفاده کرده‌اند. نتایج شبیه‌سازی آن‌ها برتری روش پیشنهادی نسبت به سایر الگوریتم‌های مشابه را نشان می‌دهد.

اباذری و همکاران [۲۳] دو هدف زمان اتمام کار و اهداف امنیتی را به‌عنوان محدودیت درنظر گرفته و با استفاده از الگوریتم‌های تکاملی روشی برای زمان‌بندی گردش کارها ارائه داده‌اند. همچنین آن‌ها در روش پیشنهادی‌شان یک رویکرد پاسخ به حمله برای کاهش برخی تهدیدات امنیتی در ابر ارائه داده‌اند. نتایج شبیه‌سازی با گردش کارهای واقعی نشان داده است که در مقایسه با سایر الگوریتم‌های موجود، راه‌حل پیشنهادی‌شان می‌تواند امنیت کلی سیستم را از نظر کیفیت امنیت و ریسک امنیتی تحت طیف گسترده‌ای از ویژگی‌های بار کاری بهبود بخشد.

لاکرا و یاداو [۲۴] از الگوریتم مرتب‌سازی نامغلوب برای زمان‌بندی گردش کارها و تخصیص منابع استفاده کرده‌اند. آنها در روش پیشنهادی‌شان دو هدف کاهش زمان اجرا و افزایش کیفیت خدمات را درنظر گرفتند. نتایج گزارش‌شده نشان از برتری روش پیشنهادی در کاهش زمان اجرا را دارد.

ژانگ و همکاران [۲۵] ابتدا یک روش بهینه‌سازی ترتیبی چندهدفه پیشنهاد داده‌اند. سپس از همین الگوریتم برای زمان‌بندی گردش کارها استفاده کرده‌اند. نتایج شبیه‌سازی گزارش‌شده نشان‌دهنده کاهش سربار محاسباتی زمان‌بندی گردش کارهای واقعی می‌باشد.

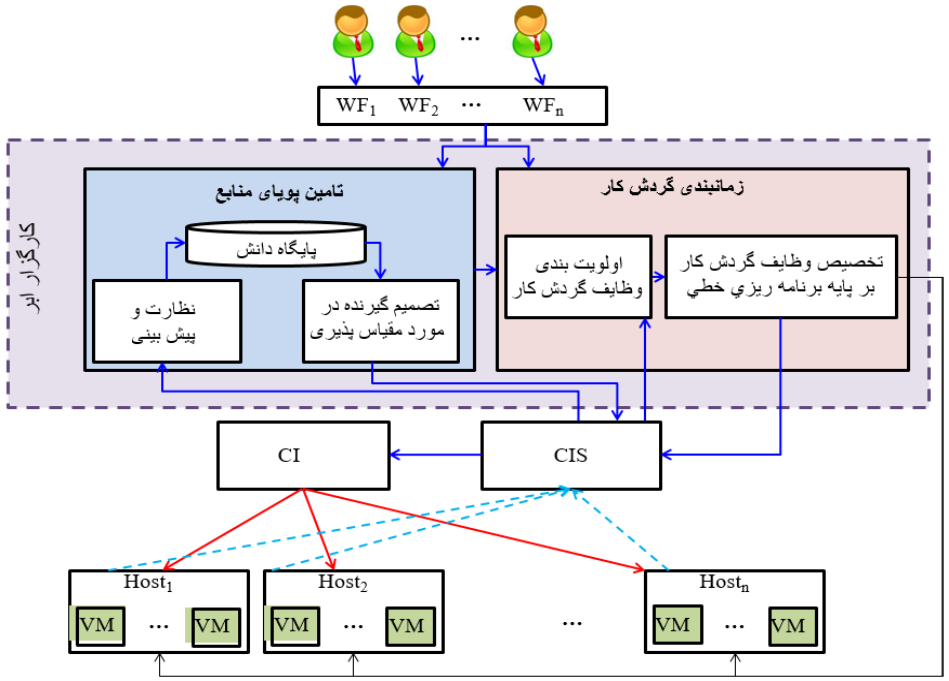
در مرجع [۲۶] یک روش مبتنی بر الگوریتم ژنتیک چندهدفه برای زمان‌بندی گردش‌کار در محیط محاسبات ابری با در نظر گرفتن دو هدف زمان تکمیل کلی و مصرف انرژی پیشنهاد شده است. در این الگوریتم یک اپراتور جهش و تقاطع جدید برای حفظ اکتشاف جمعیت در نظر گرفته شده‌اند. راهکار پیشنهادی آن‌ها یک تعادلی بین اکتشاف و بهره‌برداری در الگوریتم ژنتیک اعمال می‌کند ولی فقط دو هدف بهینه‌سازی را بهبود داده است و برای مسائل چندین هدفه راهکاری ندارد.

در مرجع [۲۷] یک رویکرد ترکیبی مبتنی بر کلونی زنبور عسل و استراتژی تسلط پارتو برای زمان‌بندی برنامه‌های گردش کار با درنظر گرفتن الزامات مختلف کیفیت خدمات در رایانش ابری ارائه شده است. برای بهبود فرایند جستجوی محلی عملگر شیفت دایره‌ای اعمال شد و سپس عملگر جهش بر روی منابع غذایی جمعیت بهبود داده شده است. کاهش زمان تکمیل کلی و هزینه پردازش در نتایج پایانی نشان داده شده است درحالی‌که روش مذکور میزان استفاده از منابع را نیز حداکثر می‌کند.

بیشتر تحقیقات فوق صرفاً بر روی زمان‌بندی گردش کارها یا تأمین منبع پویا تمرکز کرده‌اند و تعداد مقالات کمتری هردو بخش را همزمان درنظر گرفته‌اند، یا چنانچه هر دو بخش همزمان درنظر گرفته شده باشند، تعداد بسیار محدودی از پارامترهای کارایی بهبود داده شده است. در این مقاله یک روش زمان‌بندی گردش کار چندهدفه به همراه یک روش تأمین منبع پویا ارائه شده است و بهبود پارامترهایی مثل هزینه، زمان پاسخ و میزان استفاده از منابع درنظر گرفته شده و سعی می‌کند نیازهای کاربر در سطح توافقات سطح خدمات را برآورده سازد.

روش پیشنهادی

چارچوب پیشنهادی تأمین منبع فعالانه مبتنی بر پیشگویی برای زمان‌بندی گردش کار چند هدفه در رایانش ابری در شکل ۱ نشان داده شده است. در چارچوب پیشنهادی درخواست کاربران به‌صورت گردش‌کار است که شامل یک مجموعه از وظایف جزئی به هم وابسته می‌باشد. وظایف گردش کار در اختیار کارگزار ابر قرار می‌گیرند. کارگزار ابر دارای دو بخش مهم تأمین پویای منابع و زمان‌بندی وظایف گردش کار می‌باشد که به‌صورت خودمختار انجام می‌شوند. پس از تصمیم‌گیری در مورد تأمین پویای منابع و زمان‌بندی وظایف گردش کار دستورات لازم به واحد CIS سپرده می‌شود تا به واحد زیرساخت ابر (CI) تغییرات را اعلام کند تا انجام گردد. در ادامه جزئیات بیشتر دو واحد مهم تأمین پویای منابع و زمان‌بندی وظایف گردش کار در چارچوب پیشنهادی ارائه می‌شود.



شکل ۱. چارچوب پیشنهادی تأمین منبع فعالانه مبتنی بر پیشگویی برای زمانبندی گردش کار چند هدفه در رایانش ابری.

واحد تأمین پویای منابع در کارگزار ابر

در شکل ۱ نشان داده شده است که در ساختار تأمین پویای منابع سه بخش مهم وجود دارد: بخش نظارت و پیش‌بینی، بخش تصمیم‌گیرنده در مورد مقیاس‌پذیری و پایگاه دانش. به‌طور کلی، اطلاعات درخواست‌ها در دور قبلی اجرا و میزان بار کاری فعلی به‌عنوان عوامل کارایی سیستم در هر مرحله توسط بخش نظارت جمع‌آوری می‌شوند. سپس میزان بارکاری آینده با استفاده از شبکه عصبی LVQ پیش‌بینی می‌شود و میزان تخطی از SLA نیز محاسبه می‌گردد و در پایگاه دانش قرار می‌گیرند. بخش تصمیم‌گیرنده در مورد مقیاس‌پذیری با استفاده از داده‌های پایگاه دانش شامل میزان تخطی از SLA و بار کاری پیش‌بینی‌شده، با استفاده از ساختار فازی تصمیم‌گیری می‌کند که چه تعداد ماشین مجازی جدید به/از سیستم اضافه یا حذف کند که اصطلاحاً به این موضوع مقیاس‌بندی افقی گفته می‌شود. پس از این مرحله، تصمیم در اختیار CIS قرار می‌گیرد تا این واحد با ارسال دستورات مقیاس‌بندی به CI موجب اعمال تغییرات در زیرساخت ابر شود. اجزای واحد تأمین پویای منابع در کارگزار ابر در ادامه بیشتر شرح داده می‌شوند.

واحد نظارت و پیش‌بینی

این بخش در بازه‌های زمانی یکسان وضعیت سیستم را کنترل می‌کند و متغیرهای موردنیاز خود را برداشت و آنها را در یک پایگاه دانش ثبت می‌کند. یکی از مهم‌ترین بخش‌های واحد نظارت و پیش‌بینی، تحلیل داده‌ها می‌باشد. در قسمت تحلیل داده‌ها دو مورد مهم بررسی می‌گردد: مورد اول میزان تخطی از شرایط موردقبول خدمات و مورد دوم پیش‌بینی بار کاری آتی است. در صورتی که زمان پاسخگویی از زمان مجاز پاسخگویی به درخواست بیشتر باشد تخطی از شرایط سرویس رخ می‌دهد. می توان از رابطه ۱ میزان تخطی از سرویس را در مورد زمان پاسخگویی را محاسبه کرد:

$$SLA\_Violation = Re\,sponseTime - DeadlineTime \quad (۱)$$

$$SLAV\_Count_R = Count(SLAV_R > 0) \quad (۲)$$

که در رابطه فوق *ResponseTime* زمان پاسخگویی به درخواست و *DeadlineTime* زمان مجاز پاسخگویی به درخواست است و $SLAV\_Count_R$ تعداد حالتی را نشان می‌دهد که تخطی از شرایط سرویس در مورد زمان پاسخگویی رخ می‌دهد. به همین ترتیب در صورتی که هزینه پاسخگویی به درخواست از هزینه درنظر گرفته شده برای پاسخگویی به درخواست توسط کاربر بیشتر باشد تخطی از شرایط سرویس در مورد هزینه رخ می‌دهد که رابطه ۳ نحوه محاسبه آن را نشان می‌دهد.

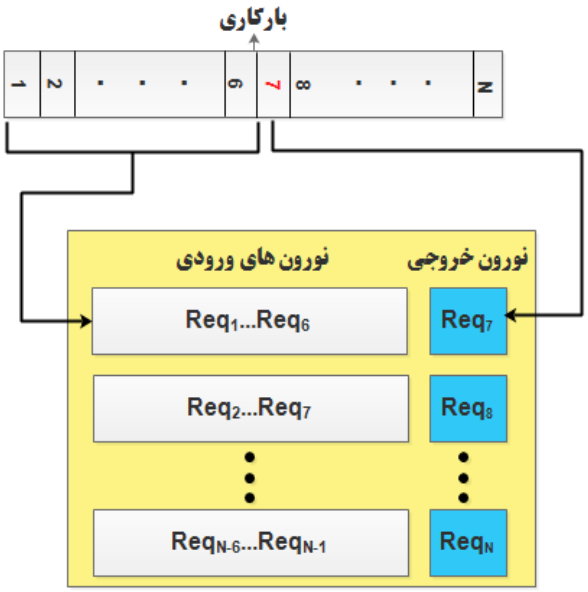
$$SLAV_{Cost} = Cost_{Re\,s} - Cost_{Re\,q} \quad (۳)$$

$$SLAV\_Count_c = Count(SLAV_{Cost} > 0) \quad (۴)$$

که در رابطه فوق، $Cost_{Req}$ هزینه ابتدایی درنظر گرفته شده برای درخواست و $Cost_{Res}$ هزینه نهایی پاسخگویی به درخواست است و $SLAV\_Count_C$ تعداد حالتی را نشان می‌دهد که تخطی از شرایط سرویس در مورد هزینه رخداده است.

به هرمیزان پیش‌بینی در این بخش دقیق‌تر باشد، تصمیم‌گیری بخش تصمیم‌گیرنده در مورد مقیاس‌پذیری، کیفیت بالاتری خواهد داشت. در روش پیشنهادی برای پیش‌بینی بار کاری از شبکه عصبی بردار چندی ساز یادگیر (LVQ) استفاده می‌شود [۲۸]. ورودی شبکه عصبی LVQ یک سری زمانی از تعداد درخواست‌های کاربر است که به صورت بارکاری وارد سیستم می‌شوند.

سری زمانی در ساختار پیشنهادی متشکل از دو آرایه ورودی و هدف است. به‌طور مشخص آرایه ورودی وضعیت حال و آرایه هدف وضعیت آینده بار کاری سیستم را مشخص می‌کند. یکی از مهم‌ترین نکات در ساخت سری زمانی ساخت دقیق این دو آرایه است. به هر میزان برش‌های زمانی دقیق‌تری در مورد این دو آرایه انجام شود، دقت تشخیص و پیش‌بینی در مورد بار کاری بالا می‌رود. در صورتی که تعداد عناصر آرایه ورودی عدد k باشد، می‌توان تعداد عناصر آرایه خروجی را در حد k/2 درنظر گرفت. نکته مهم دیگر در این زمینه این است که عناصری که در یک بازه زمانی به عنوان خروجی استفاده شده‌اند، در بازه زمانی بعدی به‌عنوان ورودی درنظر گرفته می‌شوند. این امر موجب افزایش دقت پیش‌بینی می‌شود اما مشخص است که این روش تعداد داده مورداستفاده در سری زمانی را افزایش می‌دهد و به تناسب، کمی سرعت پردازش را کاهش می‌دهد. در روش پیشنهادی به‌ازای هر ۶ واحد زمانی یک خروجی در نظر گرفته می‌شود و با توجه به توضیحات ارائه‌شده همین عمل در مورد بازه‌های بعدی نیز انجام می‌گیرد. نحوه ساخت سری زمانی از تعداد درخواست‌های کاربر به‌صورت نشان داده‌شده در شکل ۲ می‌باشد.



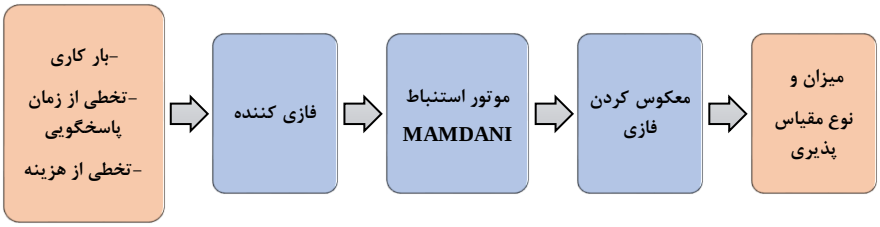
شکل ۲. ساخت سری زمانی.

روش ساخت آرایه ورودی و هدف در سری زمانی به این صورت است که هر ۶ واحد بار کاری در آرایه ورودی سری زمانی قرار می‌گیرد و واحد کار بعدی به‌عنوان خروجی یا پیش‌بینی وضعیت بعد است. به همین ترتیب یک واحد در بار کاری پیش‌آمده و عمل قبلی تکرار می‌شود.

واحد تصمیم‌گیرنده در مورد مقیاس‌پذیری

با دریافت میزان تخطی از خدمات (شامل دو مورد هزینه و زمان پاسخگویی) و پیش‌بینی بار کاری آتی، وظیفه این بخش تصمیم‌گیری مناسب برای تأمین منبع است. در این مقاله، تصمیم‌گیرنده در مورد مقیاس‌پذیری از ساختار فازی استفاده می‌کند. منطق فازی به‌منظور طراحی یک سیستم فازی از پایگاه قواعد استفاده می‌کند [۲۹]. پایگاه قواعد به مجموعه «اگر- آن‌گاه» فازی گفته می‌شود که قلب سیستم استنتاج فازی را تشکیل می‌دهد. دو روش عمده برای تعیین قواعد فازی وجود دارد: استفاده از دانش خبره و استفاده از آموزش‌های خودسازمانده مانند الگوریتم‌های شبکه عصبی که در اینجا از روش اول برای تعیین قواعد فازی استفاده شده است. به‌طور کلی ساختار یک سیستم کنترل فازی کامل از بلوک‌های فازی‌کننده، موتور استنباط، معکوس‌کننده فازی تشکیل شده است.

شکل ۳ ساختار تصمیم‌گیرنده در مورد مقیاس‌پذیری در روش پیشنهادی را نشان می‌دهد. به‌طور کلی ماژول فازی‌کننده، مقادیر اولیه ورودی‌های کنترلی را به مقادیر فازی تبدیل می‌کند. در روش پیشنهادی میزان تخطی از سرویس در زمینه زمان پاسخ، میزان تخطی از سرویس در زمینه هزینه و بار کاری پیش‌بینی‌شده به‌عنوان ورودی‌های تصمیم‌گیرنده هستند. در این مرحله، به‌واسطه موتور استنباط ممدانی تصمیم‌گیری می‌شود که براساس داده‌های ورودی چه نوع مقیاس‌پذیری انجام شود.



شکل ۳. ساختار کنترل‌کننده منطق فازی در روش پیشنهادی.

سیستم استنتاج فازی، قوانین فازی را پردازش می‌کند تا نیاز به منابع در آینده را براساس نوع بار کاری پیش‌بینی‌شده، تخطی از سرویس زمان پاسخ و تخطی از سرویس هزینه پیش‌بینی کند. در روش پیشنهادی از افراد خبره خواسته شده تا نوع و میزان مقیاس‌پذیری را تعیین کنند. برای مثال به مدیران سیستم‌ها مجموعه‌ای از قوانین «اگر- آن‌گاه» داده شده تا تعداد منابع موردنیاز برای حفظ رضایت مشتری در یک سطح قابل‌قبول را تعیین کنند. سؤالاتی که به افراد خبره داده شده است تا پاسخ و دانش آنها را استخراج کنند به صورت زیر می‌باشد:

اگر (بار کاری ..... باشد و تخطی از زمان پاسخ سرویس ..... باشد و تخطی از سرویس در زمینه هزینه ..... باشد)، آن‌گاه (مقدار مقیاس‌پذیری برابر است با .....).

ذکر این نکته ضروری است که در روش میدانی، خروجی به شکل فازی تعریف می‌شود. با توجه به متغیرهای ورودی قوانین موجب تولید یک متغیر خروجی فازی خواهند شد که در نهایت این امر پس از معکوس کردن فازی، عددی برای اولویت خود تولید می‌کند. خروجی این بخش شامل نوع مقیاس‌بندی و تعداد پیشنهادی برای افزایش، کاهش یا بدون تغییر است. برای مثال اگر (بار کاری زیاد باشد و تخطی از زمان پاسخ سرویس خیلی بد باشد و تخطی از سرویس در زمینه هزینه خیلی بد باشد)، آن‌گاه (مقدار مقیاس‌پذیری برابر است با دو)، یعنی در این شرایط دو ماشین مجازی به تعداد ماشین‌های مجازی فعلی اضافه می‌شود. پس از تصمیم‌گیری در مورد تأمین منبع در بخش تصمیم‌گیرنده، تصمیم نهایی در اختیار CIS قرار می‌گیرد تا این واحد با ارسال مقیاس‌بندی به CI موجب اعمال تغییرات در زیرساخت ابر شود.

واحد زمان‌بندی وظایف گردش کار در کارگزار ابر

در ساختار زمان‌بندی گردش کار در چارچوب روش پیشنهادی، دو بخش مهم اولویت‌بندی وظایف گردش کار و بخش تخصیص وظایف گردش کار به منابع ابر بر پایه برنامه‌ریزی خطی وجود دارد.

در ابتدا در واحد اولویت‌بندی، وظایف گردش کار براساس زمان ورود و زمان پاسخ موردنیاز اولویت‌بندی می‌شوند و براساس اولویت در صف وظایف گردش کار قرار می‌گیرند. سپس در اختیار واحد تخصیص منبع قرار می‌گیرند. واحد تخصیص منبع براساس چهار معیار پهنای باند، مقدار MIPS، میانگین زمان پاسخگویی و هزینه بر پایه برنامه‌ریزی خطی منبع مناسب را برای دسته‌ای از وظایف گردش کار اختصاص می‌دهد که جزئیات بیشتر در ادامه شرح داده می‌شوند.

واحد اولویت‌بندی وظایف گردش کار

این واحد برای مشخص کردن اولویت وظایف از رابطه ۵ استفاده می‌کند.

$$Pr\, i_i = \alpha \times \frac{1}{(Current - Time_i)} + \beta \times \frac{1}{DeadlineTime_i} \qquad (۵)$$

به‌طوری‌که در رابطه فوق، $Current$ زمان جاری، $Time_i$ زمان ورود درخواست و $DeadlineTime_i$ زمان حد مجاز پاسخگویی به درخواست می‌باشد. α و β هم مقدار وزن درنظر گرفته شده برای هریک از پارامترها می‌باشد که یک عدد بین صفر و یک است و نشان‌دهنده میزان اهمیت پارامتر مربوطه می‌باشد.

واحد تخصیص وظایف گردش کار بر پایه برنامه‌ریزی خطی

وظیفه این بخش، اختصاص منبع به وظایف گردش کار در یک واحد زمانی براساس اطلاعات دریافت‌شده از CIS و صف وظایف گردش کار اولویت‌بندی شده است.

در روش پیشنهادی الگوریتم برنامه‌ریزی خطی چندهدفه [۳۰; ۳۱]، برای تخصیص ماشین‌های مجازی متناسب با درخواست کاربر استفاده می‌شود. مسئله برنامه‌ریزی خطی، یک مسئله برنامه‌نویسی ریاضی است که در آن هدف به‌صورت $C_1X_1+C_2X_2+\ldots+C_nX_n$ است که باید مینیمم یا ماکزیمم شود و محدودیت‌ها به صورت تابع خطی از x_iها به صورت نشان داده‌شده در زیر می‌باشد.

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2+\ldots+a_{1n}x_n < b_1 \\ &\ldots \\ &a_{m1}x_1 + a_{m2}x_2+\ldots+a_{mn}x_n < b_m \end{aligned} \qquad (۶)$$

انتخاب یا انتخاب نکردن ویژگی یک ماشین مجازی با متغیر X_i مشخص می‌گردد که در صورت انتخاب، مقدار ۱ و در صورت انتخاب نکردن، مقدار ۰ می‌پذیرد. به‌طور کلی دو نوع معیار کیفیت برای ارائه خدمات ماشین‌های مجازی درنظر گرفته می‌شود. معیارهای مثبت شامل میزان پردازنده و میزان پهنای باند و معیارهای منفی شامل هزینه و زمان پاسخگویی می‌باشند. به‌طور مشخص با کاهش مقدار معیارهای منفی و افزایش مقدار معیارهای مثبت کیفیت خدمات‌رسانی به کاربران افزایش می‌یابد و به تناسب مقدار تخطی از شرایط خدمات کاهش می‌یابد. به همین دلیل در این مقاله برای الگوریتم برنامه‌ریزی خطی چندهدفه دو هدف برای بیشینه‌سازی معیارهای مثبت و دو هدف برای کمینه‌سازی معیارهای منفی درنظرگرفته می‌شود.

رابطه‌های (۷) تا (۱۰) اهداف اصلی شامل افزایش پهنای باند و قدرت پردازنده ماشین‌های مجازی و کاهش هزینه و زمان پاسخگویی آن‌ها را نشان می‌دهند. همچنین رابطه‌های (۱۱) و (۱۲) محدودیت‌های مطرح‌شده در برنامه‌ریزی خطی روی میانگین هزینه و زمان پاسخگویی را نشان می‌دهند که باید کوچک‌تر یا مساوی با شرایط درخواست‌شده کاربر باشد.

$$Maximize: \quad Z_1 = \sum_{i=1}^{S} x_i \times VM_i\_BW \tag{۷}$$

$$Maximize: \quad Z_2 = \sum_{i=1}^{S} x_i \times VM_i\_MIPS \tag{۸}$$

$$Minimize: \quad Z_3 = \sum_{i=1}^{S} x_i \times VM_i\_Cost \tag{۹}$$

$$Minimize: \quad Z_4 = \sum_{i=1}^{S} x_i \times VM_i\_Resp \tag{۱۰}$$

$$\frac{\sum_{i=1}^{S} x_i \times VM_i\_Cost}{S} \leq Req\_Cost \tag{۱۱}$$

$$\frac{\sum_{i=1}^{S} x_i \times VM_i\_Resp}{S} \leq Req\_Resp \tag{۱۲}$$

که در رابطه فوق *Req_Cost* میزان هزینه و *Req_Resp* زمان پاسخ است که توسط کاربر تعیین می‌گردند.

ارزیابی

تنظیمات اولیه آزمایش‌ها

در این مقاله به‌منظور شبیه‌سازی روش پیشنهادی در محیط ابر از ابزار CloudSim 3.0.3 [۳۲] استفاده شده است. به دلیل اینکه ابزار CloudSim با زبان جاوا نوشته شده است، محیط توسعه NetBeans به‌عنوان محیط برنامه نویسی جاوا انتخاب شده است. ساختار مرکز داده مورداستفاده در شبیه‌سازی در جدول ۱ مشخص شده است.

جدول ۱. مشخصات مراکز داده‌ها.

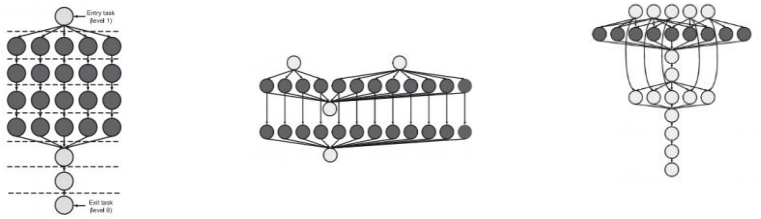
معماری	سیستم عامل	مدیریت ماشین‌های مجازی
X64	Cloud Linux	XEN

در مرکزداده، ۵ میزبان و در هر میزبان ۲۰ ماشین مجازی وجود دارد. جدول ۲ مشخصات میزبان‌ها را نشان می‌دهد. به‌طور کلی به هر میزان سخت‌افزار میزبان قوی‌تر و از سطح بالاتری برخوردار باشد میزان هزینه دسترسی به منابع مربوط به ماشین مجازی موجود در میزبان افزایش می‌یابد.

جدول ۲. مشخصات میزبان.

پهنای باند	حافظه اصلی (GB)	فرکانس (MIPS)
۱ Gbit/s	۶۴	۴۰۹۶

در شبیه‌سازی روش پیشنهادی از سه ساختار علمی مشهور گردش کار با نام Montage، Cybershake و Epigenomics به صورت نشان داده‌شده در شکل ۴ استفاده شده است. درواقع هر درخواست در قالب یکی از این ساختارهای گردش کار به سیستم وارد می‌شود.



الف) بار کاری Montage **ب) بار کاری CyberShake** **ج) بار کاری Epigenomics**

شکل ۴. نمایی از چهار ساختار علمی مشهور گردش کار مورداستفاده.

معیارهای ارزیابی

معیارهای استفاده شده برای بررسی روش پیشنهادی شامل میزان بهره‌وری، میانگین زمان پاسخ و مجموع هزینه است که در ادامه هریک شرح داده می‌شوند.

بهره‌وری: میزان بهره‌وری از رابطه ۱۳ به‌دست می‌آید [۳۳]:

$$U = \frac{AL\_MIPS}{AV\_MIPS} \qquad (۱۳)$$

که در آن AL_MIPS مقدار توان پردازشی موردنیاز و AV_MIPS مقدار توان پردازشی در دسترس و U بهره‌وری ماشین مجازی را نشان می‌دهد.

هزینه: مجموع هزینه اختصاص ماشین مجازی و هزینه ناشی از جریمه به‌ازای رخداد تخطی از شرایط سرویس را میزان هزینه می‌گویند [۳۴]. برای محاسبه میزان هزینه از رابطه ۱۴ می‌توان استفاده کرد:

$$TotalCost = VMCoct + PenaltyCost \qquad (۱۴)$$

که در رابطه فوق *VMcost* هزینه یک ماشین مجازی است که از رابطه ۱۵ به‌دست می‌آید و *PenaltyCost* هزینه جریمه‌ای است که فراهم‌کننده به‌ازای رخداد تخطی از شرایط سرویس در اختیار کاربر قرار می‌دهد و از رابطه ۱۶ به‌دست می‌آید:

$$VMCost = \sum_{i=1}^{M} VM\ Pr\ i\ ce_i \times (VM\_hour_i + VM\_Init_i) \quad (۱۵)$$

که در رابطه فوق VM_hour_i تعداد ساعت برای پاسخگویی به درخواست i ام، $VMPrice_i$ قیمت ماشین مجازی درنظر گرفته‌شده برای درخواست i ام و VM_Init_i هزینه تنظیمات اولیه ماشین مجازی است.

$$PenaltyCost = \sum_{i=1}^{M} T_i.\alpha Penalty \times T_i.Penalty \times SLAV_i \quad (۱۶)$$

که در رابطه فوق $T_i.Penalty$ نرخ جریمه و α ضریب افزایش جریمه به‌ازای تکرار رخداد است و $SLAV_i$ میزان تخطی از شرایط سرویس است که حاصل تفاوت زمان پاسخگویی به درخواست i ام و زمان موردانتظار کاربر است.

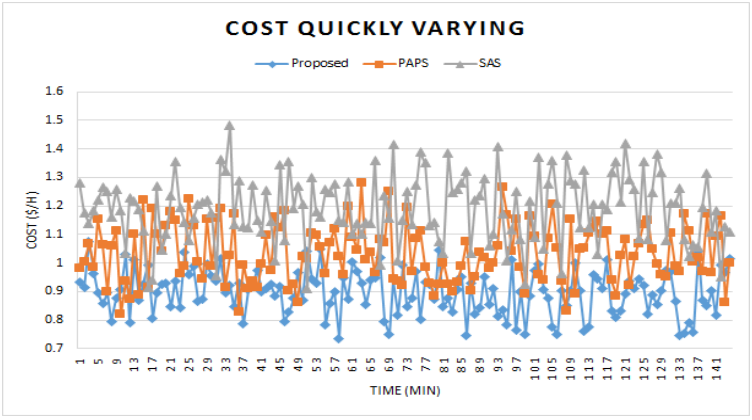
زمان پاسخگویی: تفاوت زمانی دقیق بین زمان درخواست کار و زمان تحویل کار انجام‌شده به کاربر می‌باشد [۳۵].

ارزیابی آزمایش‌ها

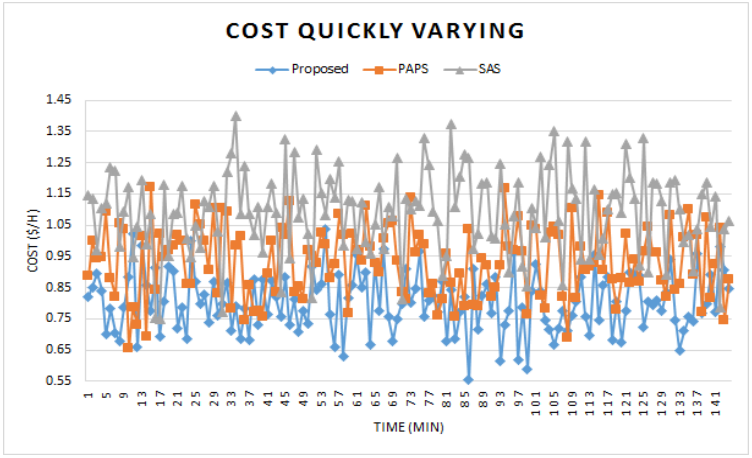
برای ارزیابی عملکرد روش پیشنهادی، سه معیار مهم برای ارزیابی درنظر گرفته شده است که عبارتند از: بهره‌وری، میانگین زمان پاسخ و مجموع هزینه. علاوه بر این، استراتژی پیشنهادی در آزمایش‌ها با الگوریتم‌های PAPS [۲۱] و SAS [۳۶] مقایسه و ارزیابی می‌شود. الگوریتم PAPS براساس اولویت‌های بودجه کاربر و محدویت زمان پاسخگویی عمل می‌کند. در آن اولویت‌بندی وظایف با تصمیم‌گیر شرطی انجام می‌شود و در بخش تأمین منبع از ساختار Best Fit بر پایه قابلیت‌های ماشین مجازی استفاده می‌کند. در الگوریتم SAS برای تأمین منبع از ساختار بردار بار استفاده می‌کند. بدین ترتیب که هر وظیفه به برداری تبدیل می‌گردد که مشخص‌کننده تعداد ماشین‌های مجازی موردنیاز است. این ساختار با استفاده از تصمیم‌گیر شرطی در مورد تأمین منابع تصمیم‌گیری می‌کند. به‌طور کلی بر پایه کاهش هزینه و محدویت زمان پاسخگویی می‌باشد و ساختار زمان‌بندی وظایف آن بر پایه انتخاب زودترین مهلت زمانی می‌باشد.

سناریو اول (مقایسه معیار کارایی هزینه)

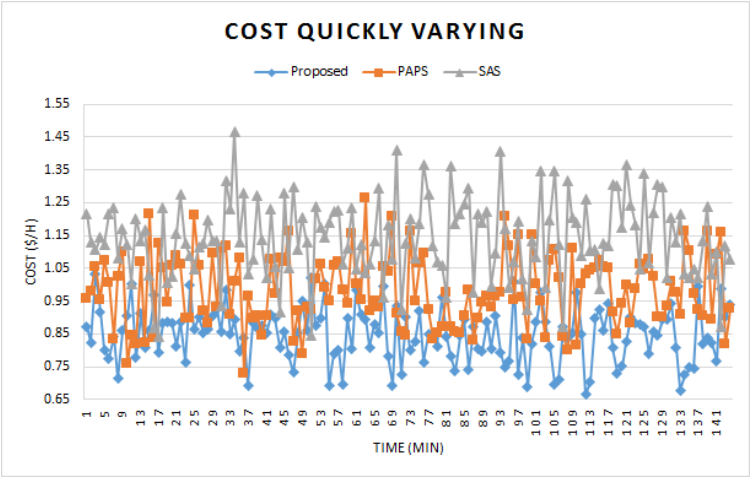
در این آزمایش میزان هزینه در روش پیشنهادی با دو الگوریتم PAPS و SCS بررسی می‌شود. مهم‌ترین شاخص در مقایسه عملکرد الگوریتم تأمین منابع، میزان هزینه است. شکل ۵ میزان هزینه روش پیشنهادی در سه ساختار گردش کار Montage، Cybershake و Epigenomics را در مقایسه با دو الگوریتم PAPS و SCS نشان می‌دهد. با توجه به نتایج حاصل شده، استفاده از برنامه‌ریزی خطی و ایجاد قیود هزینه برای تخصیص مناسب منبع و همچنین استفاده از تابع هدف برای کمینه کردن هزینه، موجب اولویت‌دهی بیشتر به هزینه روش پیشنهادی در تخصیص منبع خواهد شد. به‌طوری‌که روش پیشنهادی در تمامی ساختارهای گردش کاری مختلف عملکرد بهتری نسبت به روش‌های مورد مقایسه دارد. اما مهم‌ترین دلیل در کاهش هزینه استفاده از سیستم فازی مناسب برای تأمین منبع برپایه تخطی از شرایط سرویس به‌خصوص هزینه است که این امر موجب افزایش دقت در اضافه و کم کردن ماشین‌های مجازی نیز می‌شود.



الف) مقایسه میزان هزینه برای بار کاری Montage.



ب) مقایسه میزان هزینه برای بار کاری Cybershake.

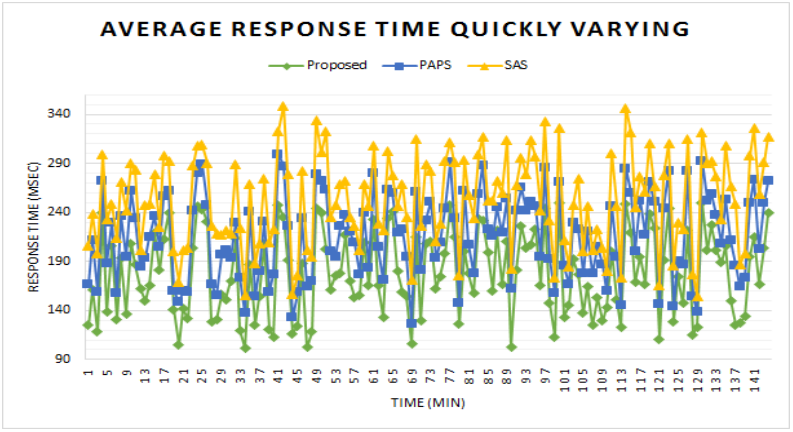


ج) مقایسه میزان هزینه برای بار کاری **Epigenomics**

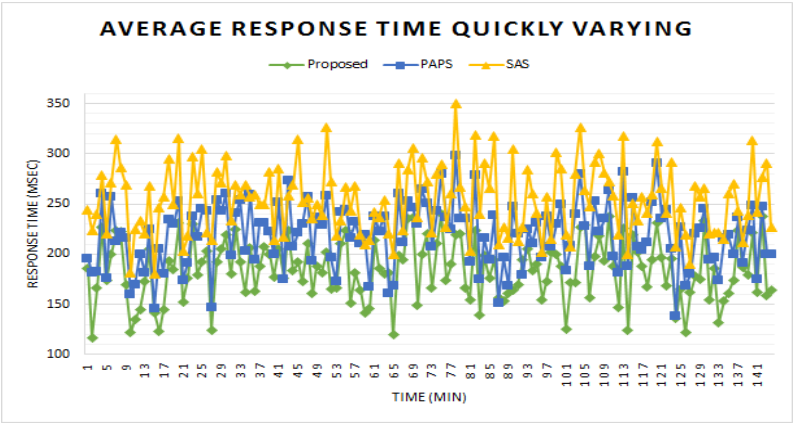
شکل ۵. مقایسه میزان هزینه روش پیشنهادی با دو روش PAPS و SCS در ساختار گردش کار مختلف.

سناریو دوم (مقایسه زمان پاسخگویی)

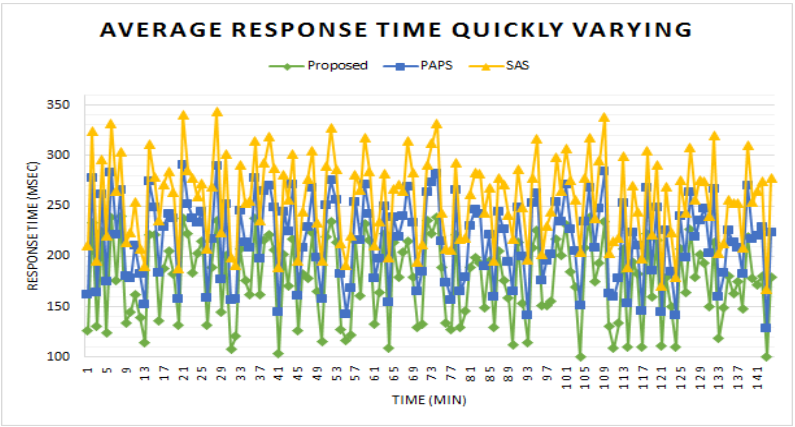
زمان پاسخگویی یکی از مهم‌ترین اهداف توافقات سطح سرویس است که نقش مؤثری در انتخاب ماشین مجازی دارد. در صورتی که زمان پاسخگویی موردنظر درخواست (حد مجاز زمان پاسخگویی) حاصل نشود، باید مقیاس‌بندی صحیح انجام شود. شکل ۶ میانگین زمان پاسخگویی روش پیشنهادی در چهار ساختار گردش کار Montage، Cybershake، Epigenomics و LIGO را در مقایسه با دو الگوریتمPAPS و SCS نشان می‌دهد.



الف) مقایسه میزان میانگین زمان پاسخگویی برای بار کاری **Montage**.



ب) مقایسه میزان میانگین زمان پاسخگویی برای بار کاری Cybershake.



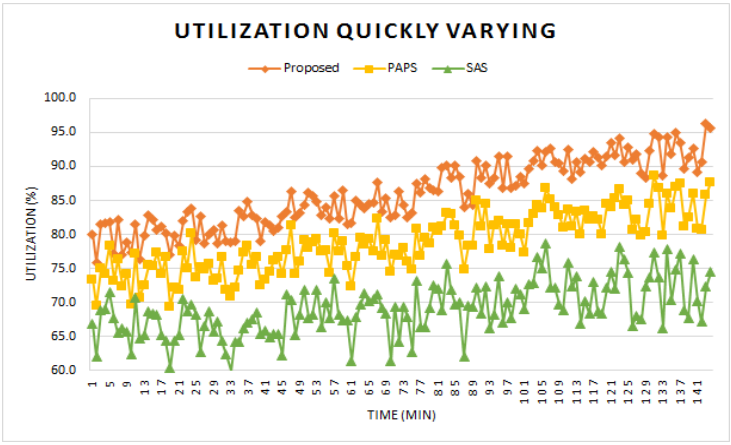
ج) مقایسه میزان میانگین زمان پاسخگویی برای بار کاری Epigenomics.

شکل ۶. مقایسه میزان میانگین زمان پاسخگویی روش پیشنهادی با دو روش PAPS و SCS در ساختار گردش کار مختلف.

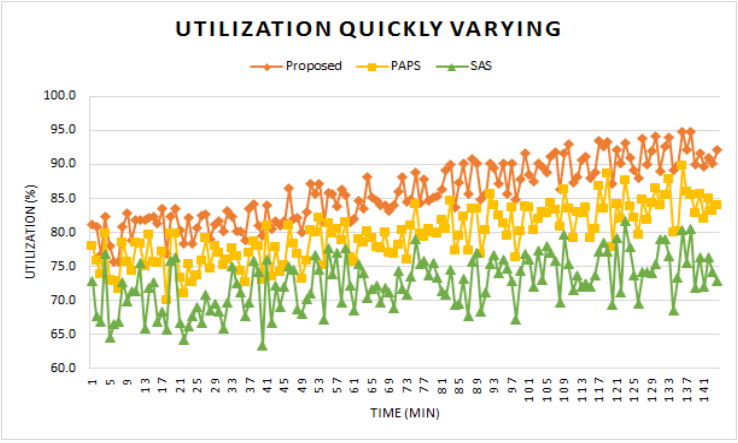
نتایج به‌دست‌آمده مشخص می‌کند که روش پیشنهادی عملکرد مطلوبی در مورد زمان پاسخگویی دارد. نکته مهم در این راستا اولویت‌بندی وظایف برحسب زمان موردنیاز برای پاسخگویی است. در بخش تأمین منبع روش پیشنهادی نیز به دلیل وجود تصمیم‌گیر فازی و استفاده از دو متغیر ورودی مهم که به‌ترتیب تخطی از زمان پاسخگویی و بار کاری پیشگویی شده می‌باشد، بهبود تأمین منبع و دسترسی دقیق به منابع حاصل می‌شود. این درحالی است که اگر تأمین منبع مناسب انجام نشود درخواستی در انتظار خواهد بود که بدون منبع می‌ماند و زمان پاسخگویی به درخواست افزایش می‌یابد و به تناسب تخطی از شرایط پذیرش سرویس پیش می‌آید. همچنین عملکرد مطلوب برنامه‌ریزی خطی به‌عنوان تشخیص پیش‌نیاز و شروط تصمیم‌گیر سیستم در بخش تخصیص منبع روش پیشنهادی موجب انتخاب دقیق‌تر منبع می‌گردد و موجب افزایش سرعت پاسخگویی نسبت به سایر روش‌های مورد مقایسه می‌شود.

سناریو سوم (مقایسه میزان بهره‌وری)

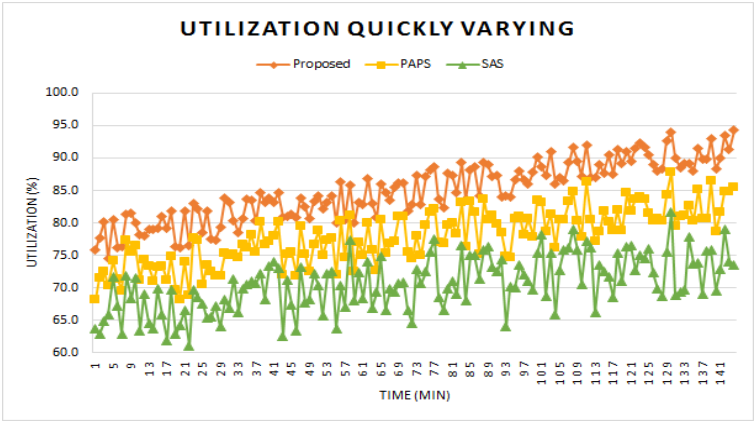
شکل ۷ میانگین بهره‌وری روش پیشنهادی در سه ساختار گردش کار Montage، Cybershake و Epigenomics را در مقایسه با دو الگوریتم PAPS و SCS نشان می‌دهد.



الف) مقایسه میزان بهره‌وری برای بار کاری Montage.



ب) مقایسه میزان بهره‌وری برای بار کاری Cybershake.



ج) مقایسه میزان بهره‌وری برای بار کاری Epigenomics

شکل ۷. مقایسه میزان بهره‌وری روش پیشنهادی با دو روش PAPS و SCS در ساختار گردش کار مختلف.

نتایج به‌دست‌آمده مشخص می‌کند که روش پیشنهادی عملکرد بهتری در مورد میزان بهره‌وری در مقایسه با دو روش دیگر دارد. نظارت دقیق بر وضعیت عملکرد منابع در ابر و تشخیص مستمر بار کاری عامل بسیار مؤثری برای کنترل وضعیت منابع در روش پیشنهادی است. درواقع به دلیل نظارت بر وضعیت CIS و استفاده از تصمیم‌گیر فازی برای تأمین منبع بر پایه نیاز کاربر، مقیاس‌بندی به نحو مطلوب انجام می‌گیرد. همچنین درنظرگیری معیار میزان قدرت پردازنده در تخصیص منبع نیز می‌تواند تأثیر به‌سزایی در افزایش بهره‌وری داشته باشد.

نتیجه‌گیری

در این مقاله رویکردی برای زمان‌بندی وظایف گردش کار بر پایه تأمین خودکار منابع ارائه شده است که با درنظر گرفتن تمامی عوامل مؤثر، مقیاس‌بندی خودکار در ابر را انجام می‌دهد. در روش پیشنهادی، درخواست کاربران به صورت گردش کار به وظایف جزئی تبدیل می‌شود. وظایف در اختیار واسط ابر قرار می‌گیرد. واسط ابر دارای دو بخش مهم برای تأمین پویای منبع و زمان‌بندی گردش کار است. پس از تصمیم‌گیری در مورد تأمین منابع دستور تغییر به واحد CIS سپرده می‌شود تا این واحد به CI تغییرات را اعلام کند تا انجام گردد. واحد زمان‌بند هم با دریافت اطلاعات از CIS پس از اولویت‌بندی وظایف گردش کار تخصیص منابع را انجام می‌دهد. روش پیشنهادی بر روی سه ساختار گردش کار مختلف آزمایش و ارزیابی شد. نتایج شبیه‌سازی نشان می‌دهد که روش پیشنهادی در مقایسه با کارهای قبلی توانسته است بهره‌وری را به‌طور متوسط به میزان ۶/۵ درصد افزایش دهد و زمان پاسخگویی و هزینه را به‌طور متوسط به‌ترتیب ۸/۳ درصد و ۷/۹ درصد کاهش دهد. در آینده تلاش می‌شود تا برای افزایش دقت تأمین منبع و جلوگیری از تخطی از شرایط سرویس سایر روش‌ها مانند ساختارهای چندمعیاره به‌جای استفاده از تصمیم‌گیر فازی بررسی شوند.

References

[1] Parida, B. R., Rath, A. K., & Swagatika, S. (2021). Load Balancing of Tasks in Cloud Computing Using Fault-Tolerant Honey Bee Foraging Approach. In D. Mishra, R. Buyya, P. Mohapatra, & S. Patnaik (Eds.), *Intelligent and Cloud Computing*. Springer Singapore. https://doi.org/10.1007/978-981-15-6202-0_6

[2] Balla, H. A., Sheng, C. G., & Jing, W. (2021). Reliability-aware: task scheduling in cloud computing using multi-agent reinforcement learning algorithm and neural fitted Q. *The International Arab Journal of Information Technology,*, *18*(1), 36-47. https://doi.org/10.34028/iajit/18/1/5

[3] Wu, L., Garg, S. K., Versteeg, S., & Buyya, R. (2014). SLA-Based Resource Provisioning for Hosted Software-as-a-Service Applications in Cloud Computing Environments. *IEEE Transactions on Services Computing*, *7*(3), 465-485. https://doi.org/10.1109/TSC.2013.49

[4] Liu, H. (2022). Research on cloud computing adaptive task scheduling based on ant colony algorithm. *Optik*, *258*, 168677. https://doi.org/10.1016/j.ijleo.2022.168677

[5] Dubey, K., Kumar, M., & Sharma, S. C. (2018). Modified HEFT Algorithm for Task Scheduling in Cloud Environment. *Procedia Computer Science*, *125*, 725-732. https://doi.org/10.1016/j.procs.2017.12.093

[6] Fakhfakh, F., Kacem, H. H., & Kacem, A. H. (2014, September 1-2). *Workflow Scheduling in Cloud Computing: A Survey*. 2014 IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations, Ulm, Germany. https://doi.org/10.1109/EDOCW.2014.61

[7] Huang, K. C., Tsai, Y. L., & Liu, H. C. (2015). Task ranking and allocation in list-based workflow scheduling on parallel computing platform. *The Journal of Supercomputing*, *71*(1), 217-240. https://doi.org/10.1007/s11227-014-1294-7

[8] Jena, R. K. (2017). Energy Efficient Task Scheduling in Cloud Environment. *Energy Procedia*, *141*, 222-227. https://doi.org/10.1016/j.egypro.2017.11.096

[9] Kaur, N., & Singh, S. (2016). A Budget-constrained Time and Reliability Optimization BAT Algorithm for Scheduling Workflow Applications in Clouds. *Procedia Computer Science*, *98*, 199-204. https://doi.org/10.1016/j.procs.2016.09.032

[10] Li, H., Ge, S., & Zhang, L. (2014, May 31 June 2). *A QoS-based scheduling algorithm for instance-intensive workflows in cloud environment.* The 26th Chinese Control and Decision Conference (2014 CCDC), Changsha, China. https://doi.org/10.1109/CCDC.2014.6852898

[11] Li, J., Su, S., Cheng, X., Huang, Q., & Zhang, Z. (2011, September 2-4). *Cost-Conscious Scheduling for Large Graph Processing in the Cloud.* 2011 IEEE International Conference on High Performance Computing and Communications, Banff, AB, Canada. https://doi.org/10.1109/HPCC.2011.147

[12] Wang, X., Wang, Y., & Zhu, H. (2012). Energy-Efficient Multi-Job Scheduling Model for Cloud Computing and Its Genetic Algorithm. *Mathematical Problems in Engineering*, *2012*, 1-16. https://doi.org/10.1155/2012/589243

[13] Safari, M., & Khorsand, R. (2018). PL-DVFS: combining Power-aware List-based scheduling algorithm with DVFS technique for real-time tasks in Cloud Computing. *The Journal of Supercomputing*, *74*(10), 5578-5600. https://doi.org/10.1007/s11227-018-2498-z

[14] Ergu, D., Kou, G., Peng, Y., Shi, Y., & Shi, Y. (2013). The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment. *The Journal of Supercomputing*, *64*(3), 835-848. https://doi.org/10.1007/s11227-011-0625-1

[15] Kong, X., Lin, C., Jiang, Y., Yan, W., & Chu, X. (2011). Efficient dynamic task scheduling in virtualized data centers with fuzzy prediction. *Journal of Network and Computer Applications*, *34*(4), 1068-1077. https://doi.org/10.1016/j.jnca.2010.06.001

[16] Khorsand, R., Safi-Esfahani, F., Nematbakhsh, N., & Mohsenzade, M. (2017). ATSDS: adaptive two-stage deadline-constrained workflow scheduling considering run-time

circumstances in cloud computing environments. *The Journal of Supercomputing, 73*(6), 2430-2455. https://doi.org/10.1007/s11227-016-1928-z

[17] Alaei, M., Khorsand, R., & Ramezanpour, M. (2021). An adaptive fault detector strategy for scientific workflow scheduling based on improved differential evolution algorithm in cloud. *Applied Soft Computing, 99*(6), 106895. https://doi.org/10.1016/j.asoc.2020.106895

[18] Paknejad, P., Khorsand, R., & Ramezanpour, M. (2021). Chaotic improved PICEA-g-based multi-objective optimization for workflow scheduling in cloud environment. *Future Generation Computer Systems, 117*(10), 12-28. https://doi.org/10.1016/j.future.2020.11.002

[19] Bahrpeyma, F., Haghighi, H., & Zakerolhosseini, A. (2015). An adaptive RL based approach for dynamic resource provisioning in Cloud virtualized data centers. *Computing, 97*(12), 1209-1234. https://doi.org/10.1007/s00607-015-0455-8

[20] Jamshidi, P., Ahmad, A., & Pahl, C. (2014, June 2-3). *Autonomic resource provisioning for cloud-based software*. Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, Hyderabad, India. https://doi.org/10.1145/2593929.2593940

[21] Shi, J., Luo, J., Dong, F., Zhang, J., & Zhang, J. (2016). Elastic resource provisioning for scientific workflow scheduling in cloud under budget and deadline constraints. *Cluster Computing*, *19*(1), 167-182. https://doi.org/10.1007/s10586-015-0530-0

[22] Belgacem, A., & Beghdad-Bey, K. (2022). Multi-objective workflow scheduling in cloud computing: trade-off between makespan and cost. *Cluster Computing, 25*(1), 579-595. https://doi.org/10.1007/s10586-021-03432-y

[23] Abazari, F., Analoui, M., Takabi, H., & Fu, S. (2019). MOWS: Multi-objective workflow scheduling in cloud computing based on heuristic algorithm. *Simulation Modelling Practice and Theory*, *93*, 119-132. https://doi.org/10.1016/j.simpat.2018.10.004

[24] Lakra, A. V., & Yadav, D. K. (2015). Multi-Objective Tasks Scheduling Algorithm for Cloud Computing Throughput Optimization. *Procedia Computer Science*, *48*, 107-113. https://doi.org/10.1016/j.procs.2015.04.158

[25] Zhang, F., Cao, J., Li, K., Khan, S. U., & Hwang, K. (2014). Multi-objective scheduling of many tasks in cloud platforms. *Future Generation Computer Systems, 37*, 309-320. https://doi.org/10.1016/j.future.2013.09.006

[26] Xia, X., Qiu, H., Xu, X., & Zhang, Y. (2022). Multi-objective workflow scheduling based on genetic algorithm in cloud environment. *Information Sciences*, *606*, 38-59. https://doi.org/10.1016/j.ins.2022.05.053

[27] Zeedan, M., Attiya, G., & El-Fishawy, N. (2023). Enhanced hybrid multi-objective workflow scheduling approach based artificial bee colony in cloud computing. *Computing*, *105*(1), 217-247. https://doi.org/10.1007/s00607-022-01116-y

[28] Sato, A., & Yamada, K. (1995). Generalized learning vector quantization. *Advances in neural information processing systems*, *8*, 423-429. https://proceedings.neurips.cc/paper/1995/file/9c3b1830513cc3b8fc4b76635d32e692-Paper.pdf

[29] Klir, G. J., & Yuan, B. (1996). *Fuzzy sets and fuzzy logic: theory and applications*. Prentice Hall. https://pubs.acs.org/doi/pdf/10.1021/ci950144a

[30] Benayoun, R., De Montgolfier, J., Tergny, J., & Laritchev, O. (1971). Linear programming with multiple objective functions: Step method (stem). *Mathematical Programming, 1*(1), 366-375. https://doi.org/10.1007/BF01584098

[31] Dantzig, G. (2016). *Linear programming and extensions*. Princeton university press. https://www.amazon.de/-/en/George-Dantzig-ebook/dp/B07ZJ2NYT6

[32] Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. F., & Buyya, R. (2011). CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, *41*(1), 23-50. https://doi.org/10.1002/spe.995

[33] Rajaee, M. (2020). Analysis of Quality of Service (QoS) Parameters of Voice over IP (VoIP). *Karafan Quarterly Scientific Journal*, *17*(1), 43-58. https://doi.org/10.48301/kssa.2020.112756

[34] Ayoubi, M., Ramezanpour, M., & Khorsand, R. (2021). An autonomous IoT service placement methodology in fog computing. *Software: Practice and Experience*, *51*(5), 1097-1120. https://doi.org/10.1002/spe.2939

[35] Karimi, H. (2021). Sensor Node Clustering Algorithm with Respect to Node Density in Wireless Sensor Networks. *Karafan Quarterly Scientific Journal*, *18*(3), 253-272. https://doi.org/10.48301/kssa.2021.269713.1360

[36] Kumar, M., & Sharma, S. C. (2018). Deadline constrained based dynamic load balancing algorithm with elasticity in cloud environment. *Computers & Electrical Engineering*, *69*, 395-411. https://doi.org/10.1016/j.compeleceng.2017.11.018