Islamic Azad University , Shiraz Branch
نشریه تحلیل مدارها، داده ها و سامانه ها
Journal of Circuits, Data and Systems Analysis
sanad.iau.ir/journal/jcdsa

Location-Aware Task Scheduling in Three-Layer IoT Systems Using Reinforcement Learning

Mohammad Zarei[1], SeyedEbrahim Dashti[2*]

[1] Department of Computer Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran
mohammad.zr1378@gmail.com
[2] Department of Computer Engineering, Jahrom Branch, Islamic Azad University, Shiraz, Iran.
sayed.dashty@gmail.com

Abstract: The rapid expansion of Internet of Things (IoT) devices has introduced critical challenges in task scheduling and resource allocation across cloud–fog ecosystems. Traditional approaches, such as location-aware scheduling and heuristic algorithms, often fail to adapt to dynamic IoT environments, where workload intensity, network traffic, and energy constraints fluctuate unpredictably. To address these issues, this study proposes a Reinforcement Learning (RL)-based scheduling framework employing a Deep Q-Network (DQN) for real-time decision-making. The model incorporates system state parameters—including task characteristics, available resources, and network conditions—into its reward function, designed to minimize task completion time, reduce congestion, and optimize energy consumption. Comparative simulation results demonstrate that the proposed RL framework significantly outperforms Genetic Algorithm (GA) and Location-Aware methods, achieving higher throughput and fewer SLA violations. Statistical testing (t-tests, p-values < 0.001) further confirms the robustness and significance of these improvements. The proposed approach thus establishes a scalable and intelligent foundation for future fog–cloud–IoT task orchestration.

Keywords: Internet of Things (IoT), Cloud Computing, Fog Computing, Location-Aware Scheduling, Reinforcement Learning

JCDSA, Vol. 3, No. 2, Summer 2025	Online ISSN: 2981-1295	Journal Homepage: https://sanad.iau.ir/en/Journal/jcdsa
Received: 2025-01-05	Accepted: 2025-08-29	Published: 2025-09-21

CITATION

Zarei, M., Dashti, SE., " Location-Aware Task Scheduling in Three-Layer IoT Systems Using Reinforcement Learning ", Journal of Circuits, Data and Systems Analysis (JCDSA), Vol. 3, No. 2, pp. 73-87, 2025.
DOI: 00.00000/0000

COPYRIGHTS

©2025 by the authors. Published by the Islamic Azad University Shiraz Branch. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 International (CC BY 4.0)
https://creativecommons.org/licenses/by/4.0

* Corresponding author

Extended Abstract

1- Introduction

The growing number of IoT devices has led to massive data generation and stringent service requirements, creating scheduling and resource management challenges in distributed fog–cloud environments [1,2]. Key issues such as communication delay, high energy consumption, and heterogeneous resources limit performance [3]. Traditional heuristics, including genetic algorithms [4], ant colony optimization [5], and location-aware strategies [6], are insufficiently adaptive to real-time variations [7]. This study proposes a Reinforcement Learning (RL)-based adaptive scheduling model, leveraging Deep Q-Networks (DQN) to autonomously learn optimal task allocation under dynamic workloads [8].

2- Methodology

The scheduling problem is formulated as a Markov Decision Process (MDP), where system states represent task requirements, resource availability, and network status. RL agents are distributed across IoT, Fog, and Cloud layers, each learning local policies while maintaining global coordination [6,9]. The DQN approximates Q-values using neural networks, with experience replay and ε-greedy exploration improving convergence stability. The reward function penalizes SLA violations, delays, and excessive energy use, while rewarding efficient, localized execution [7]. Complexity analysis shows polynomial overhead during training but negligible cost during inference, making the approach scalable for real-time IoT workloads [10].

3- Results and discussion

Simulations were conducted in Python using a dataset of approximately 3,000 IoT tasks with metadata such as task size, deadline, and SLA constraints. The RL-based scheduler was benchmarked against GA and Location-Aware methods. Results show:

- Throughput: RL reached 0.98 tasks/sec, surpassing GA (0.82) and Location-Aware.
- SLA Violation Rate: RL reduced violations to 1.7%, compared to GA's 8.2% and Location-Aware's 5.3%.
- Execution Time: RL consistently outperformed baselines.

To confirm significance, Welch's independent t-tests were performed: RL vs GA ($t = -32.04$, $p \approx 4.55\times10^{-182}$) and RL vs Location-Aware ($t = -19.45$, $p \approx 2.50\times10^{-77}$). Both p-values < 0.001 validate statistical reliability [9]. These results demonstrate that RL can dynamically adapt to workload intensity, heterogeneous resources, and fluctuating network conditions, outperforming static heuristic methods [11].

4- Conclusion

This study introduced a reinforcement learning-based scheduling framework for fog–cloud–IoT environments. By employing DQN-based agents, the model autonomously optimizes task completion time, minimizes SLA violations, and improves throughput [6,9]. Contributions include:

- A distributed RL architecture adaptive to dynamic IoT workloads,
- Integration of SLA-awareness and energy efficiency into scheduling decisions [7],
- Empirical improvements over GA and Location-Aware methods validated through rigorous statistical testing [9],
- Open dataset and source code to ensure reproducibility and future extensions [10].

This research provides a scalable, intelligent, and adaptive foundation for future IoT–fog–cloud systems, paving the way for federated RL across fog nodes and security-aware scheduling in large-scale deployments [12].

Islamic Azad University , Shiraz Branch
نشریه تحلیل مدارها، داده ها و سامانه ها
Journal of Circuits, Data and Systems Analysis
sanad.iau.ir/journal/jcdsa

زمان بندی مکان آگاه در سیستم‌های اینترنت اشیا سه لایه با کمک یادگیری تقویتی

محمد زارعی[۱]، سید ابراهیم دشتی[۲*]

۱- دانشکده برق و کامپیوتر، واحد شیراز، دانشگاه آزاد اسلامی (Mohammad.zr1378@gmail.com)
۲- دانشکده برق و کامپیوتر، واحد جهرم، دانشگاه آزاد اسلامی (Sayed.Dashty@gmail.Com)

چکیده: گسترش سریع دستگاه‌های اینترنت اشیا چالش‌های قابل توجهی در برنامه‌ریزی وظایف و تخصیص منابع در محیط‌های ابر-مه-اینترنت اشیا ایجاد کرده است. الگوریتم‌های برنامه‌ریزی سنتی، مانند الگوریتم زمان‌بندی آگاه از محل داده در ابر-مه دارای محدودیت‌های ذاتی در سازگاری و مقیاس‌پذیری در مواجهه با محیط‌های پویا و متغیر اینترنت اشیا هستند. برای غلبه بر این محدودیت‌ها، این پژوهش چارچوبی جدید مبتنی بر یادگیری تقویتی پیشنهاد می‌کند. رویکرد پیشنهادی از یک شبکه Q عمیق برای یادگیری پویا در تخصیص بهینه وظایف و استراتژی‌های استفاده از منابع در محیط ابر-مه استفاده می‌کند. روش پیشنهادی قادر به تصمیم‌گیری‌های مستقل و مبتنی بر داده برای تخصیص منابع است. وضعیت این سیستم شامل ویژگی‌های وظایف، منابع در دسترس و شرایط شبکه است، در تصمیم‌گیری‌های تخصیص وظایف، تابع پاداش به گونه‌ای طراحی شده است که زمان کلی تکمیل وظایف، تراکم شبکه و مصرف انرژی را به حداقل برساند. با اتخاذ رویکرد یادگیری تقویتی، روش پیشنهادی می‌تواند به تغییرات بار کاری، تراکم شبکه و دسترسی به منابع در زمان واقعی پاسخ دهد. برای ارزیابی عملکرد روش پیشنهادی، مجموعه‌ای جامع از آزمایش‌ها با استفاده از شبیه‌سازی سناریوهای واقعی ابر-مه انجام شد. عملکرد زمان‌بند مبتنی بر یادگیری تقویتی با الگوریتم‌های دیگر مقایسه شد. نتایج نشان می‌دهد که مدل زمان‌بندی مبتنی بر یادگیری تقویتی به‌طور قابل توجهی عملکرد بهتری از نظر کاهش زمان تکمیل وظایف، تعادل بار و کارایی شبکه دارد.

واژه های کلیدی: اینترنت اشیا، محاسبات ابری، محاسبات مه، زمان‌بندی مکان‌آگاه، یادگیری تقویتی

نوع مقاله: پژوهشی
DOI: 00.00000/0000
تاریخ ارسال مقاله: ۱۴۰۳/۱۰/۱۶ تاریخ پذیرش مقاله: ۱۴۰۴/۰۶/۰۷ تاریخ چاپ مقاله: ۱۴۰۴/۰۶/۳۱

۱- مقدمه

گسترش سریع دستگاه‌های فراگیر و اینترنت اشیا باعث ایجاد تحولی در زندگی انسان شده است. اکثر دستگاه‌ها مانند تلفن‌های هوشمند، لپ‌تاپ‌ها و دستگاه‌های پوشیدنی دارای منابع پردازشی ضعیف‌تر، اتصال شبکه محدود و حافظه و ذخیره‌سازی کمتری هستند [1, 2]. امروزه بسیاری از برنامه‌ها، از جمله برنامه‌های داده‌محور، معمولاً توسط خدمات ابری پشتیبانی می‌شوند [3, 4] . رشد سریع اینترنت اشیا و افزایش دستگاه‌های متصل، چالش‌های مهمی را در مدیریت منابع و برنامه‌ریزی وظایف[۲] در محیط‌های ابر و مه ایجاد کرده است. این چالش‌ها شامل تأخیرهای ارتباطی، تراکم شبکه و مصرف زیاد انرژی است که عملکرد سیستم‌های اینترنت اشیا را تحت تأثیر قرار می‌دهد. از دیگر چالش‌های موجود می‌توان به حجم بسیار زیاد داده‌ها اشاره کرد که در شکل (۱) شاهد رشد اطلاعات تا سال ۲۰۲۵ خواهیم بود [5]. در این میان، روش‌های سنتی زمان‌بندی وظایف، نظیر الگوریتم‌های مبتنی بر آگاهی از مکان داده‌ها [6]، به دلیل پویایی محیط‌های اینترنت اشیا و تغییرات مداوم در منابع، کارایی کافی ندارند. حجم بالای داده‌هایی که دستگاه‌های متصل تولید می‌کنند مشکلات و دشواری‌های قابل‌توجهی ایجاد می‌کند [7, 8] . شرکت بین‌المللی داده پیش‌بینی می‌کند که تا سال ۲۰۲۵، ۷۵ میلیارد شیء دارای حسگر به شبکه متصل خواهند شد و ۵۰ تریلیون دستگاه شبکه‌ای شامل ۵۰۰ میلیون حسگر، وجود خواهد داشت [9, 10] . رایانش ابری به یک انتخاب جذاب برای کاربران تبدیل شده است، زیرا از نظر هزینه برای ذخیره‌سازی و پردازش حجم زیادی از اطلاعات مقرون‌به‌صرفه است، اما همچنین محدودیت‌هایی نیز دارد [11, 12] . در این راستا، زیرساخت‌ها و مکانیزم‌های کارآمدی برای مدیریت و ذخیره‌سازی داده‌ها، منابع و انرژی مورد نیاز است. رایانش مه توسط شرکت سیسکو به‌عنوان راهکاری برای جبران محدودیت‌های ذاتی

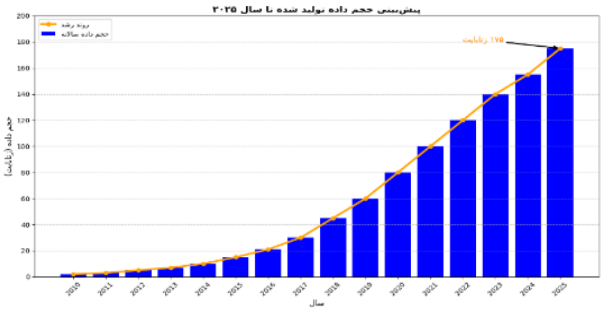
* نویسنده مسئول

[2] Makespan

رایانش ابری، مانند تأخیر بالا در ارتباطات انتها به انتها، تراکم ترافیک و هزینه‌های بالای ارتباطات پیشنهاد شده است [13, 14] . به منظور غلبه بر این چالش‌ها، روش‌های یادگیری تقویتی[1] به عنوان یک راهکار نوین برای زمان‌بندی وظایف در سیستم‌های ابر-مه معرفی شده‌اند. در این روش، عامل‌های هوشمند با تعامل مداوم با محیط و یادگیری از پاداش‌های دریافتی، قادر به اتخاذ تصمیمات بهینه در خصوص تخصیص منابع و برنامه‌ریزی وظایف می‌باشند. یادگیری تقویتی این امکان را فراهم می‌آورد که سیستم به صورت پویا و بدون نیاز به دخالت دستی، به تغییرات محیطی پاسخ داده و بهترین سیاست‌های تصمیم‌گیری را اتخاذ کند. در این پژوهش، یک چارچوب زمان‌بندی وظایف بر اساس یادگیری تقویتی برای سیستم‌های اینترنت اشیا در محیط‌های ابر-مه پیشنهاد شده است. در این چارچوب، از شبکه یادگیری Q عمیق[2] برای یادگیری پویای استراتژی‌های تخصیص منابع استفاده می‌شود. مدل یادگیری، وضعیت‌های سیستم از جمله ویژگی‌های وظایف، منابع در دسترس و شرایط شبکه را به عنوان ورودی دریافت کرده و با استفاده از تابع پاداش به گونه‌ای آموزش داده می‌شود که هدف به حداقل رساندن زمان کلی تکمیل وظایف، کاهش تراکم شبکه و بهینه‌سازی مصرف انرژی محقق گردد. مزیت اصلی این رویکرد، توانایی آن در پاسخگویی به تغییرات بار کاری، تراکم شبکه و شرایط منابع به صورت بلادرنگ[3] است. بر خلاف الگوریتم‌های سنتی، روش یادگیری تقویتی با توجه به ماهیت تطبیقی آن، قادر است در محیط‌های پویای اینترنت اشیا عملکرد بهتری داشته باشد. به منظور ارزیابی عملکرد چارچوب پیشنهادی، شبیه‌سازی‌های جامعی با استفاده از پایتون انجام شد. نتایج نشان داد که روش مبتنی بر یادگیری تقویتی در مقایسه با الگوریتم‌های مبتنی بر آگاهی از مکان داده‌ها، عملکرد بهتری داشته و در کاهش زمان تکمیل وظایف، کاهش ترافیک شبکه و بهبود مقیاس‌پذیری سیستم موثرتر بوده است.

یکی از مهم‌ترین انگیزه‌های این تحقیق، ناکارآمدی روش‌های سنتی زمان‌بندی در محیط‌های پویا و ناهمگون اینترنت اشیا است. الگوریتم‌های کلاسیک مانند روش‌های مبتنی بر آگاهی از مکان داده‌ها، در شرایطی که بارکاری و وضعیت منابع دائماً تغییر می‌کند، عملکرد پایداری ندارند. همچنین، با توجه به افزایش تعداد دستگاه‌های IoT و تولید حجم عظیم داده‌ها، نیاز به رویکردی هوشمند، بلادرنگ و انطباق‌پذیر بیش از پیش احساس می‌شود. استفاده از یادگیری تقویتی به عنوان یک راهکار هوش مصنوعی، این امکان را فراهم می‌سازد که بدون دخالت دستی، تصمیمات تخصیص منابع و زمان‌بندی وظایف به صورت بهینه و تطبیقی انجام شود. دستاوردهای اصلی این پژوهش به شرح زیر است:

- ارائه یک چارچوب زمان‌بندی وظایف مبتنی بر یادگیری تقویتی برای سیستم‌های سه‌لایه اینترنت اشیا شامل لایه IoT، مه و ابر.



شکل (۱): حجم داده تولید شده تا سال ۲۰۲۵

- طراحی ساختار لایه‌ای برای آموزش مستقل هر لایه با عامل یادگیری تقویتی مجزا و هماهنگ‌سازی بین عامل‌ها برای پاسخ به شرایط پویا.
- طراحی تابع پاداش اختصاصی برای هر لایه به منظور بهینه‌سازی هم‌زمان معیارهایی نظیر تأخیر، مصرف انرژی و تراکم شبکه.
- پیاده‌سازی و ارزیابی روش پیشنهادی و مقایسه عملکرد با الگوریتم‌های سنتی مکان‌محور.
- بهبود قابل توجه در معیارهای عملکردی نظیر زمان تکمیل وظایف، تعادل بار و کارایی منابع در مقایسه با روش‌های موجود.

ساختار مقاله به‌صورت زیر سازمان‌دهی شده است: در بخش ۲، پیشینه زمان‌بندی در معماری‌های ابر-مه-اینترنت اشیا مرور می‌شود و شکاف‌های پژوهشی با یک جدول مقایسه‌ای خلاصه شده است. بخش ۳، چارچوب پیشنهادی مبتنی بر یادگیری تقویتی شامل مدل سه‌لایه، طراحی تابع پاداش و شبه‌کُد *DQN* همراه با تحلیل پیچیدگی زمانی را تشریح می‌کند؛ در بخش ۴، محیط شبیه‌سازی، سناریوها و معیارهای ارزیابی معرفی می‌شوند و نتایج به‌همراه آزمون‌های آماری (*t-test* و *p-value*) و مقایسه با روش‌های مبنا گزارش می‌گردد. در نهایت، بخش ۵ جمع‌بندی یافته‌ها و مسیرهای پژوهشی آتی را ارائه می‌دهد.

۲- پیشینه تحقیق

در سال‌های اخیر، با رشد سریع اینترنت اشیا و سیستم‌های محاسبات ابری و مه، روش‌های مختلفی برای زمان‌بندی وظایف به منظور بهبود عملکرد سیستم‌های ابر-مه پیشنهاد شده است. الگوریتم‌های سنتی مبتنی بر روش‌های ابتکاری و فرابتکاری مانند الگوریتم‌های ژنتیک، الگوریتم‌های کلونی مورچه و سایر روش‌های هوش مصنوعی، برای بهینه‌سازی زمان‌بندی وظایف در این سیستم‌ها مورد استفاده قرار گرفته‌اند. با این حال، این الگوریتم‌ها به دلیل عدم توانایی در تطبیق با محیط‌های پویا و متغیر اینترنت اشیا، نیاز به الگوریتم‌های جدید و سازگارتر احساس می‌شود. یکی از رویکردهای نوین در این زمینه، استفاده از یادگیری تقویتی برای بهبود زمان‌بندی وظایف در محیط‌های ابر-مه است. یادگیری تقویتی به دلیل توانایی یادگیری از طریق تعامل با محیط و دریافت بازخورد، یک راهکار مناسب برای محیط‌های پویا و متغیر اینترنت اشیا به شمار می‌رود.

[1] Reinforcement Learning (RL)
[2] Deep Q-Learning Network
[3] Real-time

روش‌های زمان‌بندی مبتنی بر یادگیری تقویتی:

۱- روش‌های مبتنی برQ-Learning ساده:

یکی از روش‌های اولیه در یادگیری تقویتی برای زمان‌بندی وظایف، استفاده از Q-Learning ساده است. در این روش، یک عامل[1] به ازای هر وضعیت-عمل[2] مقادیر Q را به‌روزرسانی می‌کند. این روش اگرچه برای مسائل ساده کاربردی است، اما در مسائل پیچیده با فضای حالت بزرگ، کارایی آن کاهش می‌یابد.

۲- روش‌های مبتنی بر شبکه عصبی عمیق:

برای مقابله با محدودیت‌هایQ-Learning ، استفاده از شبکه‌های عصبی عمیق[3] به عنوان تخمین‌گر Q-Value پیشنهاد شده است. در این رویکرد، عامل یادگیری تقویتی به جای ذخیره Q-Value برای هر وضعیت-عمل، از یک شبکه عصبی برای تقریب این مقادیر استفاده می‌کند. این روش به دلیل توانایی در یادگیری روابط پیچیده و کار با فضای حالت بزرگ، به عنوان یکی از روش‌های مؤثر در زمان‌بندی وظایف در محیط‌های ابر-مه شناخته می‌شود.

۳- روش‌های مبتنی بر یادگیری تقویتی عمیق

یادگیری تقویتی عمیق (DRL) ترکیبی از یادگیری تقویتی و شبکه‌های عصبی عمیق است. این روش‌ها، مانند DQN و روش‌های مبتنی برActor-Critic، قادر به بهبود عملکرد زمان‌بندی وظایف در سیستم‌های ابر-مه هستند. یکی از مزایای این روش‌ها، توانایی آنها در یادگیری از داده‌های تاریخی و تصمیم‌گیری‌های بهینه بلادرنگ است.

۴- روش‌های داده‌محور و مکان‌محور:

در روش‌های مکان‌محور، موقعیت مکانی داده‌ها و منابع برای بهبود عملکرد در نظر گرفته می‌شود. به‌عنوان مثال، در روش‌های مبتنی بر آگاهی از مکان داده‌ها، تخصیص وظایف به منابع نزدیک‌تر انجام می‌شود تا تأخیر ارتباطی کاهش یابد. با این حال، این روش‌ها به دلیل ثابت بودن داده‌های ورودی و تغییرات محیطی، توانایی سازگاری با تغییرات سریع را ندارند. رویکردهای مبتنی بر DA[15] بر جایگذاری پایدار سرویس‌ها برای بارهای کاری اینترنت اشیا تمرکز دارند، کار ما از یادگیری تقویتی برای تطبیق پویا با تغییرات بار کاری و بهینه‌سازی تخصیص وظایف در زمان واقعی بهره می‌گیرد. همچنین DLJSF [16] با تمرکز بر نزدیکی داده‌ها موفق به کاهش زمان‌بندی وظایف شده است. چارچوب-MT OSFاز تحلیل سلسله مراتبی[4] برای اولویت‌بندی گره‌های مه استفاده می‌کند، رویکرد پیشنهادی ما با بهره‌گیری از یادگیری تقویتی، به تغییرات محیطی بلادرنگ پاسخ می‌دهد و انعطاف‌پذیری بیشتری را ارائه می‌دهد. GRASP[5] در زمان‌بندی وظایف IoT به‌دنبال تعادل میان مصرف انرژی و رعایت مهلت زمانی است. در سال‌های اخیر، با افزایش پیچیدگی محیط‌های رایانش مه و ابر، پژوهش‌ها به سمت روش‌های هوشمند و به‌ویژه یادگیری تقویتی و زمان‌بندی آگاه از داده‌ها سوق یافته‌اند [16] .در مدلDLJSF ، زمان‌بندی آگاه از مکان را با هدف بهبود نزدیکی داده‌ها و کاهش تأخیر معرفی کرد. همچنین، [17] مروری جامع بر روش‌های ابتکاری در زمان‌بندی وظایف در محیط‌های IoTارائه کرده است [20] .از الگوریتم یادگیری تقویتی چندهدفه برای متعادل‌سازی بار در معماری لبه–مه–ابر استفاده کرده که نتایج آن نشان‌دهنده‌ی بهبود در بهره‌وری منابع و کاهش تأخیر است. پژوهش [19] با استفاده از روش GRASP نیز توانسته است عملکرد بهتری نسبت به روش‌های کلاسیک ارائه دهد. این روند نشان می‌دهد که استفاده از الگوریتم‌های هوشمند مانند RL و متاهیورستیک‌ها به‌ویژه در سال‌های ۲۰۲۴ به بعد، جایگزینی مؤثر برای روش‌های سنتی در محیط‌های پویا شده‌اند. برای ارزیابی جایگاه پژوهش حاضر نسبت به مطالعات پیشین، مجموعه‌ای از مقالات سال ۲۰۲۴ انتخاب و در قالب جدول (۱) مقایسه شده‌اند. این مقایسه شامل هدف، تکنیک‌های مورد استفاده، ابزار شبیه‌سازی، نوع داده، معیارهای ارزیابی، مزایا و معایب، و نوآوری هر مقاله می‌باشد. هدف این جدول، مشخص نمودن خلأهای موجود در تحقیقات پیشین و موقعیت نوآورانه پژوهش حاضر است.

۳- روش پیشنهادی

در این پژوهش، یک روش مبتنی بر یادگیری تقویتی برای زمان‌بندی وظایف در سیستم‌های ابر-مه پیشنهاد شده است. برخلاف روش‌های سنتی که معمولاً تمام لایه‌ها را به عنوان یک سیستم یکپارچه در نظر می‌گیرند، روش ما به گونه‌ای طراحی شده که هر لایه به‌طور جداگانه توسط الگوریتم یادگیری تقویتی مورد پردازش قرار می‌گیرد. این رویکرد امکان تصمیم‌گیری محلی برای هر لایه را فراهم می‌آورد، در حالی که در عین حال هماهنگی کلی میان لایه‌ها حفظ می‌شود.

در ادامه به بررسی چند مفهوم کلیدی می‌پردازیم:

وضعیت:

- برای هر لایه، وضعیت به صورت مستقل تعریف شده است. لایه اینترنت اشیا: وضعیت شامل ویژگی‌های دستگاه‌های اینترنت اشیا مانند ظرفیت باتری، حجم داده تولیدی و محدودیت‌های زمانی وظایف است.
- لایه مه: وضعیت شامل اطلاعات منابع موجود در گره‌های مه مانند ظرفیت پردازش، حافظه، پهنای باند و تأخیر ارتباطی است.
- لایه ابر: وضعیت شامل پارامترهایی مانند ظرفیت محاسباتی بالا، دسترسی به ذخیره‌سازی بزرگ و تأخیر شبکه می‌شود.

عملیات:

- عملیات تخصیص وظایف به منابع مختلف برای هر لایه به صورت مجزا تعریف شده است.
- لایه اینترنت اشیا: وظایف می‌توانند به گره‌های مه یا مراکز ابر منتقل شوند.

[1] Agent
[2] State-Action
[3] Deep Neural Network (DNN)

جدول (۱): مقایسه‌ای از تحقیقات پیشین

عنوان مقاله	نوآوری	پارامتر ها	مزیت ها	معایب	مجموعه داده	ابزار شبیه سازی	هدف
Resource-aware fog service placement using deferred acceptance in edge computing [15]	استفاده از الگوریتم پذیرش معوق (DA)برای جای‌گذاری سرویس‌ها در محاسبات مهی (fog computing)است که این الگوریتم با ایجاد ارتباط پایدار بین سرویس‌ها و منابع، نسبت به روش‌های سنتی بهتر عمل می‌کند و از نظر پایداری منابع و سرعت با الگوریتم‌های ژنتیکی (GA) قابل مقایسه است.	زمان راه‌اندازی تأخیر در برقراری ارتباط اولیه زمان کل اجرا	پایداری بهتر نسبت به FIFO و الگوریتم‌های ژنتیکی(GA) کاهش تأخیر نسبت به FIFO وGA عملکرد قوی در کارهای پیچیده IoT پایداری در تخصیص منابع سرعت در زمان همگرایی کارایی در تخصیص منابع و اجرای وظایف	نیاز به اجرای مجدد در محیط‌های پویا افزایش سربار در تغییرات مکرر وضعیت نیاز به بهبود در زمان راه‌اندازی	این مجموعه داده شامل بار کاری مصنوعی است که نیازهای پردازش ویدئوی بلادرنگ دوربین‌های ترافیکی در سطح شهر را نشان می‌دهد. این داده‌ها نمایانگر تقاضاهای سنگین و حساس به زمان هستند که برای نظارت ترافیک شهری استفاده می‌شوند.	بستر آزمایشی سفارشی (محیط تست اختصاصی) معماری مبتنی بر پایتون ارکستراسیون و تخصیص منابع در فضای ابری	بهینه‌سازی جای‌گذاری سرویس‌های مه برای برنامه‌های IoT حساس به تأخیر است. هدف به حداقل رساندن چالش‌هایی مانند تأخیر، تغییرپذیری منابع و تحرک است. (این مطالعه به‌طور خاص بر مدیریت هوشمند ترافیک به عنوان یک کاربرد نمونه تمرکز دارد).
DLJSF: Data-Locality Aware Job Scheduling IoT tasks in fog-cloud computing environments [16]	معرفی یک الگوریتم ابتکاری به نام DLJSFاست که تعادل بین نزدیکی داده و مهاجرت وظایف را در سیستم‌های مه-ابری برقرار می‌کند.	زمان اجرای کل (Makespan) بار شبکه (Network Load) هزینه انتقال داده (Data Transfer Costs)	کارایی بالا در شرایط مختلف دستیابی به ۸۷٪ از حل بهینهLP توزیع بار و کاهش زمان اجرای کل بهبود مداوم عملکرد در شرایط مختلف	افزایش زمان پردازش در صورت کمبود منابع گره‌های مه محدودیت وابسته به کیفیت منابع مه	این مجموعه داده شامل داده‌های مصنوعی با اندازه‌های مختلف (۲۳ مگابایت، ۱۲۸ مگابایت و ۶۷۶ مگابایت) است که برای شبیه‌سازی بارهای کاری معمول در IoT استفاده شده‌اند	CloudSim NetBeans IDE GLPK	بهینه‌سازی زمان‌بندی وظایف IoTبا در نظر گرفتن نزدیکی داده‌ها برای کاهش پهنای باند شبکه و تأخیر در محیط‌های مه-ابری
Advancements in heuristic task scheduling for IoT applications in fog-cloud computing [17]	پیشنهادهایی برای پیشرفت زمان‌بندی در محاسبات مه با استفاده از یادگیری ماشین برای انطباق بلادرنگ، یادگیری فدرال برای زمان‌بندی مشارکتی و هوش مصنوعی قابل توضیح جهت شفاف‌سازی مدل‌های ابتکاری است.	استفاده از منابع بهره‌وری انرژی زمان پاسخ‌دهی اولویت وظایف امنیت	بهبود انطباق‌پذیری و عملکرد با روش‌های ترکیبی کارایی در مصرف انرژی و مقیاس‌پذیری با روش‌های الهام‌گرفته از طبیعت و مبتنی بر یادگیری استفاده از یادگیری ماشین و یادگیری فدرال برای بهبود تصمیم‌گیری در زمان‌بندی	مشکل در مقیاس‌پذیری روش‌های حریصانه نیاز بالا به محاسبات در روش‌های فرابتکاری پیچیدگی در انطباق مدل‌های مبتنی بر یادگیری نیاز به انطباق‌پذیری بلادرنگ و شفافیت در زمان‌بندی مه چالش هایی مانند ناهمگنی سیستم، امنیت و شرایط پویا	مجموعه داده خاصی ندارد	از ابزار شبیه‌سازی خاصی استفاده نمی‌کند.	هدف تحلیل روش‌های زمان‌بندی وظایف مبتنی بر ابتکار در محاسبات مه است. این بررسی استراتژی‌هایی مانند روش‌های مبتنی بر اولویت، حریصانه، فرابتکاری، یادگیری‌محور، ترکیبی و الهام‌گرفته از طبیعت را پوشش می‌دهد. هدف اصلی بهینه‌سازی استفاده از منابع، کاهش مصرف انرژی، و افزایش کارایی اجرای وظایف IoT در محیط‌های مه است، همراه با شناسایی نقاط ضعف و پیشنهاد مسیرهای پژوهشی آینده.
Multi-Objective Task-Aware Offloading and	نوآوری این مطالعه شامل مکانیزم	مصرف انرژی زمان پاسخ‌دهی	کاهش قابل توجه زمان پاسخ‌دهی	عدم انطباق‌پذیری	این مجموعه داده‌های	iFogSim2	هدف این مطالعه بهینه‌سازی

Scheduling Framework for Internet of Things Logistics [18]	زمان‌بندی آگاه از وظیفه با استفاده از فرآیند تحلیل سلسله مراتبی (AHP)برای اولویت‌بندی گره‌های مه است. همچنین شامل مدیریت تحمل خطا برای مقابله با خرابی وظایف و گره‌ها و تخلیه وظایف بر اساس اولویت می‌شود.	نسبت خرابی وظایف معیارهای انتخاب گره مه فاصله اقلیدسی	کاهش مصرف انرژی کاهش نسبت خرابی وظایف یکپارچه‌سازی تحمل خطا انتخاب چندمعیاری گره‌های مه تخلیه وظایف بر اساس اولویت	مبتنی بر یادگیری ماشین وابستگی به الگوریتم FCFSبرای زمان‌بندی وظایف در ابر	مصنوعی مربوط به وظایف لجستیکی IoT است که با استفاده از یک فرآیند پواسون تولید شده و شامل وظایفی با اندازه‌ها و اولویت‌های مختلف است.		تخلیه و زمان‌بندی وظایف در لجستیک مبتنی بر IoT است.
A greedy randomized adaptive search procedure for scheduling IoT tasks in virtualized fog–cloud computing [19]	کاربرد الگوریتم GRASPدر VFCCکه از جستجوی تصادفی و تطبیقی استفاده می‌کند تا به صورت تکراری راه‌حل‌هایی با بهره‌وری انرژی بالا و در چارچوب مهلت زمانی پیدا می کند.	نسبت رضایت از مهلت زمانی میانگین زمان پاسخ‌دهی مصرف انرژی زمان اجرای کل (Makespan) زمان اجرا (Runtime)	عملکرد برتر در معیارهایی مانند رضایت از مهلت زمانی، مصرف انرژی، میانگین زمان پاسخ‌دهی و زمان اجرای کل ویژگی چند شروع (Multistart).	نیاز به بهبود در مدیریت زمان‌های شروع متنوع وظایف عدم تطبیق کامل با مقیاس‌های مختلف توان پردازنده (CPU) محدودیت در مدیریت مسائل امنیت و حریم خصوصی	مجموعه داده شامل داده‌های مصنوعی است که انواع مختلف وظایف IoT را شبیه‌سازی می‌کند و شامل پارامترهایی مانند اندازه وظایف، حافظه و نیازهای مهلت زمانی است.	شبیه‌سازی در این مطالعه با استفاده از مجموعه داده مصنوعی و وظایفی که از طریق یک توزیع تصادفی یکنواخت تولید شده‌اند انجام شده است. این شبیه‌سازی در یک محیط رایانه‌ای با پیکربندی‌های مشخص CPUو حافظه انجام شده است.	هدف بهینه‌سازی زمان‌بندی وظایف برای برنامه‌های IoTدر محیط محاسبات مه-ابری مجازی‌سازی شده (VFCC)است. تمرکز اصلی بر کاهش مصرف انرژی و رعایت مهلت زمانی وظایف است.
Location-aware job scheduling for IoT systems using cloud and fog [6]	نوآوری این مطالعه معرفی یک چارچوب زمان‌بندی وظایف با آگاهی از محل داده (DLSFC)است که پردازش مبتنی بر مه را برای وظایف با داده‌های سنگین در اولویت قرار می‌دهد. این چارچوب از یک مدل برنامه‌ریزی خطی استفاده می‌کند تا بهینه‌سازی پیچیده و NP-hard زمان‌بندی وظایف را در محیط‌های توزیع‌شده IoT حل کند.	تأخیر شبکه استفاده از پهنای باند مهلت‌های تکمیل وظایف هزینه‌های انتقال داده	بهره‌وری بالا بهبود عملکرد کاهش بار شبکه بهبود تأخیر	پیچیدگی افزوده در توازن پردازش وظایف محلی و مهاجرت داده‌ها نیاز به مدیریت دقیق مهاجرت داده‌ها	مجموعه داده شامل داده‌های مصنوعی است که بارهای کاری معمول در IoT با جریان‌های داده‌ای با فرکانس بالا را شبیه‌سازی می‌کند. این داده‌ها شامل وظایف با اندازه‌ها و نیازهای زمانی مختلف است تا تقاضاهای دنیای واقعی IoT را به تصویر بکشد.	شبیه‌سازی‌ها با استفاده از مدل سفارشی‌سازی‌شده‌ای انجام شده که به‌طور خاص برای زمان‌بندی وظایف در محیط‌های مه-ابری طراحی شده است.	هدف، بهینه‌سازی زمان‌بندی وظایف برای سیستم‌های IoTبا استفاده از نزدیکی داده‌ها برای افزایش کارایی در محیط‌های مه-ابری است که شامل کاهش تأخیر و استفاده از پهنای باند برای برنامه‌های IoTدر مقیاس بزرگ می‌شود و وظایف IoT که نیاز به پردازش بلادرنگ دارند را با حداقل تأخیر در انتقال داده‌ها هدف قرار می‌دهد.

- لایه مه: وظایف می‌توانند در همان گره مه پردازش شوند یا به مرکز ابر ارسال شوند.
- لایه ابر: وظایف به منابع محاسباتی با اولویت مصرف بهینه انرژی تخصیص داده می‌شوند.

تابع پاداش:

- هر لایه تابع پاداش مختص به خود را دارد که به نحوی طراحی شده تا اهداف محلی و کلی سیستم را بهینه کند:
  - لایه اینترنت اشیا: کاهش مصرف انرژی دستگاه‌های IoT و حداقل‌سازی تأخیر ارسال داده‌ها.
  - لایه مه: کاهش تراکم گره‌های مه و بهینه‌سازی پهنای باند.
  - لایه ابر: حداقل‌سازی زمان تکمیل وظایف سنگین و کاهش هزینه‌های انتقال داده.

فرآیند آموزش:

- هر لایه به صورت مستقل توسط عامل یادگیری تقویتی آموزش داده می‌شود.

تعامل عامل‌ها:

- عامل‌های مربوط به هر لایه از طریق یک مکانیزم هماهنگی با یکدیگر تعامل دارند.
- به عنوان مثال، اگر منابع مه دچار تراکم شوند، عامل لایه مه وظایف اضافی را به مرکز ابر منتقل می‌کند.

معماری سیستم:

معماری سیستم پیشنهادی شامل سه لایه اصلی است که در شکل (۲) آورده شده است:

۱. لایه دستگاه‌های اینترنت اشیا: این لایه شامل دستگاه‌های تولیدکننده داده مانند حسگرها، تلفن‌های هوشمند و دستگاه‌های پوشیدنی است. این دستگاه‌ها وظایف را تولید و برای پردازش به سیستم ارسال می‌کنند.

۲. لایه مه: در این لایه، گره‌های محاسباتی مه قرار دارند که نزدیک به دستگاه‌های اینترنت اشیا هستند. این گره‌ها خدمات محاسباتی و ذخیره‌سازی محلی را فراهم می‌کنند و تأخیر ارتباطی کمتری دارند.

۳. لایه ابر: این لایه شامل مراکز داده قدرتمند با ظرفیت‌های ذخیره‌سازی و محاسباتی بالا است. این لایه برای وظایف سنگین و منابعی که به تأخیر کمتر حساس نیستند، استفاده می‌شود.

در شکل (۲) نمای کلی از معماری سیستم آورده شده که هر کدام از اجزای آن را در ادامه بررسی خواهیم کرد.

اجزای یادگیری تقویتی

در روش پیشنهادی، یک عامل یادگیری تقویتی به عنوان هسته اصلی زمان‌بند عمل می‌کند. اجزای کلیدی این سیستم شامل موارد زیر است:

۱. وضعیت[1]: وضعیت شامل ویژگی‌های وظایف (مانند حجم داده و مهلت زمانی)، منابع در دسترس (مانند ظرفیت پردازش و حافظه) و وضعیت شبکه (مانند پهنای باند و تاخیر) است.

۲. عملیات[2]: مجموعه‌ای از عملیات تخصیص وظایف به منابع مختلف. این عملیات ممکن است شامل تخصیص وظیفه به یک گره مه یا یک مرکز ابر باشد.

۳. تابع پاداش[3]: به منظور بهینه‌سازی، تابع پاداش به گونه‌ای طراحی شده که کاهش زمان تکمیل وظایف، کاهش مصرف انرژی و کاهش تراکم شبکه را تشویق کند. به عنوان مثال، اگر یک وظیفه با موفقیت روی یک گره محلی اجرا شود، یک پاداش مثبت در نظر گرفته می‌شود، اما اگر نیاز به انتقال داده باشد، جریمه منفی به عامل تعلق می‌گیرد.

فرآیند یادگیری تقویتی:

فرآیند زمان‌بندی وظایف با استفاده از یادگیری تقویتی در مراحل زیر انجام می‌شود:

۱. مشاهده وضعیت اولیه: عامل وضعیت محیط (مانند وظایف موجود، منابع در دسترس و وضعیت شبکه) را مشاهده می‌کند.

۲. انتخاب عمل: عامل با استفاده از سیاست ε-greedy، یکی از عملیات ممکن را انتخاب می‌کند. در ابتدای یادگیری، عامل با احتمال بیشتر به اکتشاف می‌پردازد و با پیشرفت یادگیری، اکتشاف به استثمار (استفاده از بهترین تجربه‌های قبلی) تغییر می‌کند.

۳. اجرای عمل :عملیات تخصیص وظیفه به یک منبع انجام می‌شود.

۴. مشاهده وضعیت جدید و دریافت پاداش :پس از اجرای عملیات، وضعیت جدید محیط و پاداش مرتبط با آن محاسبه می‌شود.

۵. به‌روزرسانی Q-Value: با استفاده از معادله Q-Learning، Q-Valueبرای وضعیت-عمل انجام شده به‌روزرسانی می‌شود:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a))$$

α: نرخ یادگیری

γ : فاکتور تخفیف

r: پاداش دریافت شده

$Q(s',a')$: ارزش بهترین عملیات در وضعیت جدید

در ادامه به بررسی فلوچارت مربوط به یادگیری تقویتی می‌پردازیم:

اجرای کلی کد بصورت زیر است:

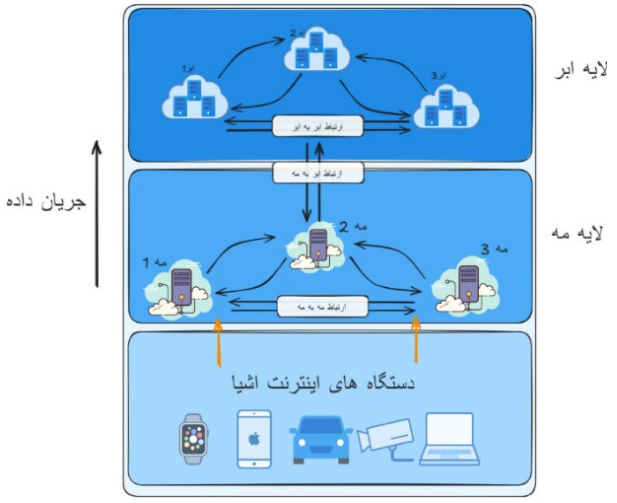
ورودی‌ها:

وظایف: مجموعه‌ای از وظایف که باید برنامه‌ریزی شوند. هر وظیفه شامل اطلاعاتی مثل حجم داده، محدودیت زمانی و مکان داده است.

منابع: مجموعه‌ای از منابع در دسترس برای اجرای وظایف. این منابع می‌توانند گره‌های مه یا ابر باشند.

پارامترهای یادگیری: شامل نرخ یادگیری(α)، نرخ تخفیف(γ) ، و نرخ کاوش(ϵ)

حداکثر تکرارها: تعداد اپیزودها و مراحل یادگیری



شکل (۲): معماری سه لایه از سیستم ابر-مه

[1] State
[2] Actions
[3] Reward

خروجی‌ها:

سیاست بهینه: جدولی که تصمیم‌گیری بهینه را نشان می‌دهد، یعنی هر وظیفه به کدام منبع تخصیص داده شود.

تخصیص وظایف: لیستی که نشان می‌دهد هر وظیفه به کدام منبع اختصاص یافته است.

کلیات عملکرد:

۱. مرحله آغازین:

در این مرحله، محیط اولیه الگوریتم آماده می‌شود:

- جدول *Q-Table* : یک ماتریس دوبعدی است که در آن ستون‌ها نشان‌دهنده‌ی اقدامات ممکن و ردیف‌ها نشان‌دهنده‌ی وضعیت‌های ممکن سیستم هستند. تمامی مقادیر اولیه جدول با صفر مقداردهی می‌شوند.
- سیاست اولیه *(π)*: سیاست تصادفی یا مشخص‌شده‌ای تعیین می‌شود که به الگوریتم اجازه می‌دهد در شروع یادگیری اقداماتی را انجام دهد.
- پارامترهای یادگیری: مقادیر اولیه برای α (نرخ یادگیری)، γ (نرخ تخفیف) و ϵ (نرخ کاوش) تنظیم می‌شود.
- وضعیت اولیه وضعیت سیستم شامل صف وظایف، منابع موجود و شرایط شبکه تعریف می‌شود.

۲. مرحله یادگیری:

در این مرحله، الگوریتم به‌صورت تدریجی و از طریق تکرار اپیزودها و مراحل، سیاست خود را بهبود می‌بخشد.

گام‌های هر اپیزود:

- مشاهده وضعیت اولیه: الگوریتم وضعیت فعلی سیستم را از محیط دریافت می‌کند. این وضعیت می‌تواند شامل تعداد وظایف در صف، میزان استفاده از منابع و تأخیر شبکه باشد.
- انتخاب اقدام: الگوریتم با استفاده از روش ϵ-*greedy* یکی از دو راه زیر را برای انتخاب اقدام انتخاب می‌کند:
- کاوش: با احتمال ϵ، یک اقدام تصادفی انتخاب می‌شود.
- بهره‌برداری: با احتمال $1-\epsilon$، اقدام با بالاترین مقدار $Q(s,a)$ از جدول انتخاب می‌شود.
- اجرای اقدام: اقدام انتخاب‌شده اجرا می‌شود؛ به این معنا که وظیفه موردنظر به منبع انتخاب‌شده تخصیص داده می‌شود. سیستم وظیفه را اجرا کرده و وضعیت جدید (s') و پاداش (r) را به الگوریتم بازمی‌گرداند
- پاداش: براساس عملکرد تخصیص محاسبه می‌شود. از عوامل موثر می‌توان به زمان تکمیل وظیفه، تأخیر شبکه و انرژی اشاره کرد.
- به‌روزرسانی جدول Q : مقدار $Q(s,a)$ براساس فرمول زیر محاسبه می‌شود که ترکیبی از پاداش فعلی و بهترین ارزش اقدام در وضعیت آینده است.

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a))$$

- به‌روزرسانی وضعیت: وضعیت فعلی (s) به وضعیت جدید (s') تغییر می‌یابد.
- بررسی معیار توقف: اگر همه وظایف تخصیص داده شده باشند یا تعداد مراحل از *MaxSteps* فراتر رود، اپیزود خاتمه می‌یابد.
- کاهش نرخ کاوش (اختیاری): پس از هر اپیزود، مقدار ϵ کاهش می‌یابد مثلاً $(\epsilon=\epsilon\times0.99\epsilon)$ تا الگوریتم در ادامه بیشتر به بهره‌برداری از سیاست بهینه بپردازد.

۳. مرحله تصمیم‌گیری:

پس از آموزش، الگوریتم سیاست بهینه را برای تخصیص وظایف به منابع جدید استفاده می‌کند

- مشاهده وضعیت: وضعیت سیستم شامل وظایف جدید و منابع موجود به الگوریتم ارائه می‌شود.
- انتخاب اقدام: برای هر وظیفه جدید، اقدام بهینه با استفاده از سیاست یادگرفته‌شده انتخاب می‌شود.
- اجرای اقدام و به‌روزرسانی وضعیت: وظیفه به منبع مربوطه تخصیص داده شده و وضعیت سیستم به‌روز می‌شود.
- بازگشت خروجی: الگوریتم تخصیص نهایی وظایف را به‌عنوان خروجی ارائه می‌دهد.

برتری‌های روش پیشنهادی

۱. **قابلیت تطبیق با محیط‌های پویا:** یادگیری تقویتی به دلیل توانایی انطباق با تغییرات محیطی در زمان واقعی، نسبت به روش‌های سنتی مانند برنامه‌ریزی خطی یا الگوریتم‌های ابتکاری عملکرد بهتری دارد.

۲. **کاهش مصرف انرژی:** عامل یادگیری وظایف را به منابع محلی تخصیص می‌دهد و نیاز به انتقال داده‌ها را کاهش می‌دهد.

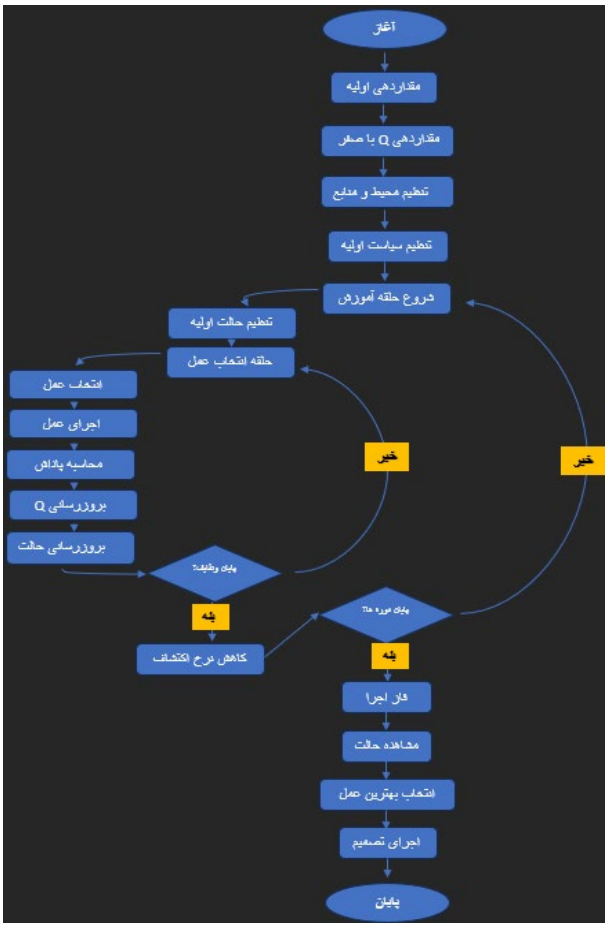
۳. **افزایش سرعت اجرا:** کاهش تأخیر ارتباطی و تخصیص هوشمند وظایف به منابع نزدیک باعث تسریع اجرا می‌شود.

۴. **قابلیت گسترش‌پذیری:** این روش به‌طور مستقیم قابلیت اجرا در سیستم‌های با مقیاس بزرگ‌تر را دارد، زیرا عامل یادگیری تقویتی می‌تواند سیاست‌های تخصیص را برای سیستم‌های پیچیده‌تر بهبود دهد.

در شکل (۳) و جدول (۲) فلوچارت و شبه کد الگوریتم یادگیری تقویتی را مشاهده می‌کنیم. در شکل (۴) نگاهی کلی به جزئیات یادگیری تقویتی داریم و با سازوکار آن آشنا می‌شویم.

تحلیل سربار محاسباتی

پیچیدگی زمانی الگوریتم یادگیری تقویتی پیشنهادی بر مبنای معماری DQN تحلیل می‌شود. فرآیند آموزش شامل تکرارهای متعددی از اپیزودهای یادگیری و به‌روزرسانی وزن‌های شبکه عصبی است.



شکل (۳) فلوچارت الگوریتم یادگیری تقویتی

جدول (۲): شبه کد الگوریتم یادگیری تقویتی

Inputs:

Task set T;
resource set R;
learning rate α;
discount factor γ;
exploration rate ε;
maximum number of episodes and steps
Output:
Optimal policy *π mapping tasks to resources
Initialization:
Initialize Q-Table with zeros; set an initial random policy; prepare environment with tasks and resources
Training Phase:
For each episode:
1. Observe initial state s
2. Select action aaa using ε-greedy policy
3. Execute action, observe next state s′ and reward r
4. Update Q-value
5. Update state s′←s
6. Optionally reduce ε
Decision Phase:
For each new task:
1. Observe current state s
2. Select action
3. Execute action and assign the task to the selected resource

با فرض تعداد اپیزودهای آموزش E، تعداد مراحل هر اپیزود T، پیچیدگی زمانی یک بار پیش‌بینی و به‌روزرسانی شبکه F، پیچیدگی زمانی الگوریتم به صورت زیر قابل بیان است:

$$Time\ Complexity = O(E \cdot T \cdot F)$$

در مدل فعلی، شبکه عصبی از چند لایه پنهان با تعداد نرون محدود استفاده می‌کند، بنابراین F معمولاً در حد $O(n^2)$ می‌باشد. به‌طور تجربی با تعیین مقدار E=۱۰۰۰ و T=۵۰، سربار محاسباتی در زمان آموزش وجود دارد، ولی این بار در فاز اجرایی (inference) بسیار کاهش می‌یابد .مزیت اصلی روش پیشنهادی، آن است که فرآیند یادگیری تنها یک‌بار و به‌صورت آفلاین انجام می‌شود، و پس از آن تصمیم‌گیری‌ها در زمان اجرای واقعی با پیچیدگی بسیار پایین انجام می‌گردد. بنابراین، الگوریتم پیشنهادی برای سیستم‌های بلادرنگ و منابع محدود نیز قابل اجرا است .شکل ۳ نمایش‌دهنده‌ی دیاگرام توالی تعامل بین مؤلفه‌های سیستم پیشنهادی است. در این دیاگرام، توالی ارسال، ثبت، تصمیم‌گیری، اجرای وظیفه و دریافت پاداش توسط عامل RL نمایش داده شده که ساختار رفتاری مدل پیشنهادی را شفاف‌تر می‌سازد.

۴- ارزیابی

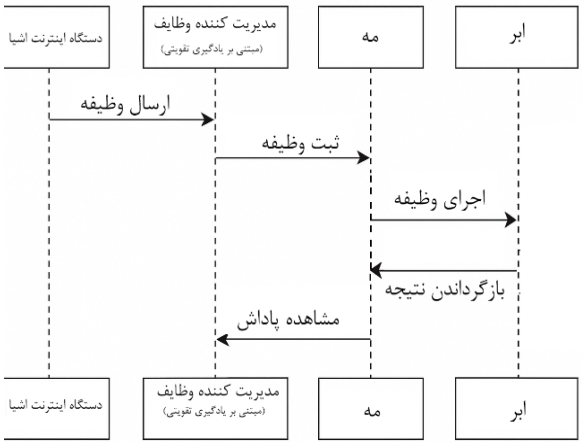
هدف ارزیابی

هدف از این ارزیابی، بررسی عملکرد الگوریتم زمان‌بندی مبتنی بر یادگیری تقویتی در مقایسه با روش‌های رایج در سیستم‌های ابر-مه است. در این ارزیابی، معیارهای کلیدی مانند مدت زمان اجرای کل، مقدار ارتباطات بین مراکز محاسباتی، الگوی انتقال داده، زمان اجرای الگوریتم، عمر شبکه، مصرف انرژی و مقیاس‌پذیری سیستم مورد بررسی قرار گرفته است.

محیط شبیه‌سازی

مشخصات‌سیستم:

سیستم ارزیابی شامل ۸ مرکز مه و ۱ مرکز ابر است که در مناطق جغرافیایی مختلف پراکنده شده‌اند. مشخصات منابع فیزیکی این مراکز بصورت کامل در جدول (۳) آورده شده است.



شکل (۴): دیاگرام توالی تعامل مولفه‌های یادگیری تقویتی

ابزارهای شبیه‌سازی :

- پایتون برای مدل‌سازی و شبیه‌سازی محیط ابر-مه.
- حل‌کننده خطی *GLPK* برای حل مسائل برنامه‌ریزی خطی.
- گوگل کولب و پایچرم برای اجرای شبیه‌سازی و پیاده‌سازی الگوریتم.

روش‌های مقایسه شده

سه رویکرد برای مقایسه در نظر گرفته شده است:

۱. روش محلی سازی: در این روش، وظایف فقط در مکان‌هایی که داده‌های مربوطه ذخیره شده‌اند اجرا می‌شوند.
۲. روش مهاجرت: در این روش، امکان انتقال داده‌ها برای بهبود کارایی در نظر گرفته می‌شود.
۳. روش یادگیری تقویتی: روش مبتنی بر یادگیری تقویتی، که هدف آن تصمیم‌گیری بهینه و پویا برای تخصیص وظایف و استفاده از منابع است.

شاخص‌ها مورد ارزیابی

۱. مدت زمان اجرای کل: زمان کلی اجرای تمام وظایف در سیستم.
هزینه ارتباطی[1]: میزان داده‌های جابه‌جا شده بین مراکز.
۲. الگوی انتقال داده[2]: میزان داده‌های محلی و غیرمحلی پردازش‌شده.
۳. مصرف انرژی[3]: کل انرژی مصرفی در مراکز ابر، مه و دستگاه‌های اینترنت اشیا.
۴. زمان اجرای الگوریتم[4]: مدت زمانی که الگوریتم برای تخصیص وظایف و اجرای آن‌ها نیاز دارد.
۵. مقیاس‌پذیری[5]: توانایی سیستم برای مدیریت بارهای کاری در حال تغییر.

تحلیل و بررسی شاخص‌های مورد نظر

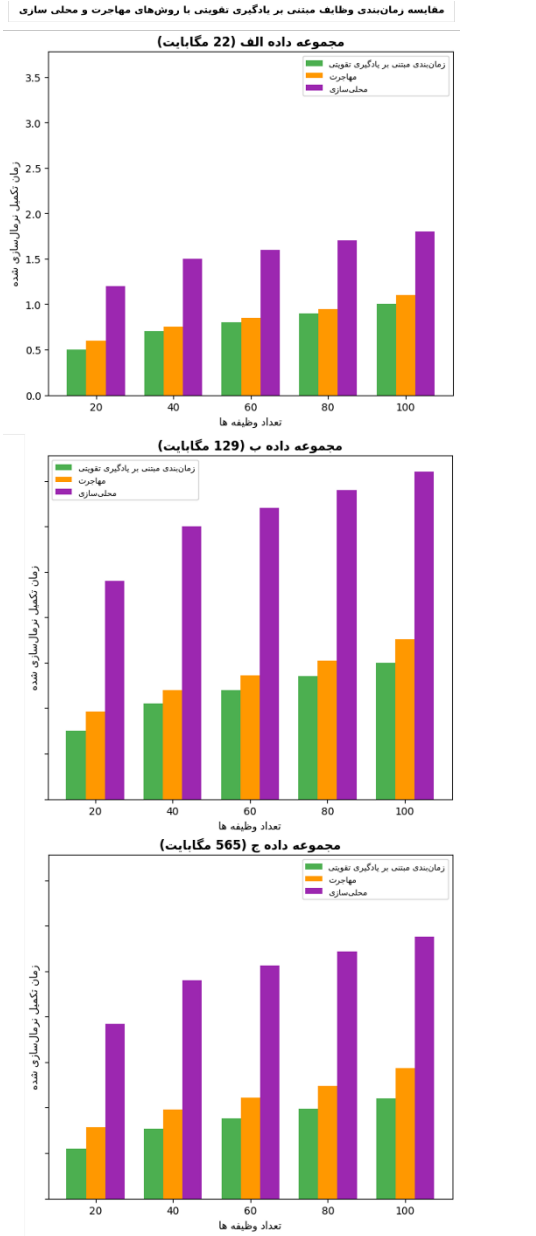
۱. کاهش مدت زمان اجرای کل

- هدف: بررسی این‌که روش پیشنهادی مبتنی بر یادگیری تقویتی چقدر در کاهش مدت زمان کلی اجرای وظایف مؤثر است.
- نتیجه: نتایج نشان می‌دهند که روش یادگیری تقویتی به‌طور متوسط ۲۰٪ سریع‌تر از روش‌های محلی سازی6 و مهاجرت۷ عمل می‌کند.
- تحلیل: دلیل این بهبود، قابلیت یادگیری پویا و تطبیق با شرایط سیستم است که امکان استفاده بهینه‌تر از منابع را فراهم می‌آورد.

در شکل (۵) مقایسه زمان تکمیل وظایف برای سه الگوریتم محلی، مهاجرت و یادگیری تقویتی را مشاهده می‌کنیم.

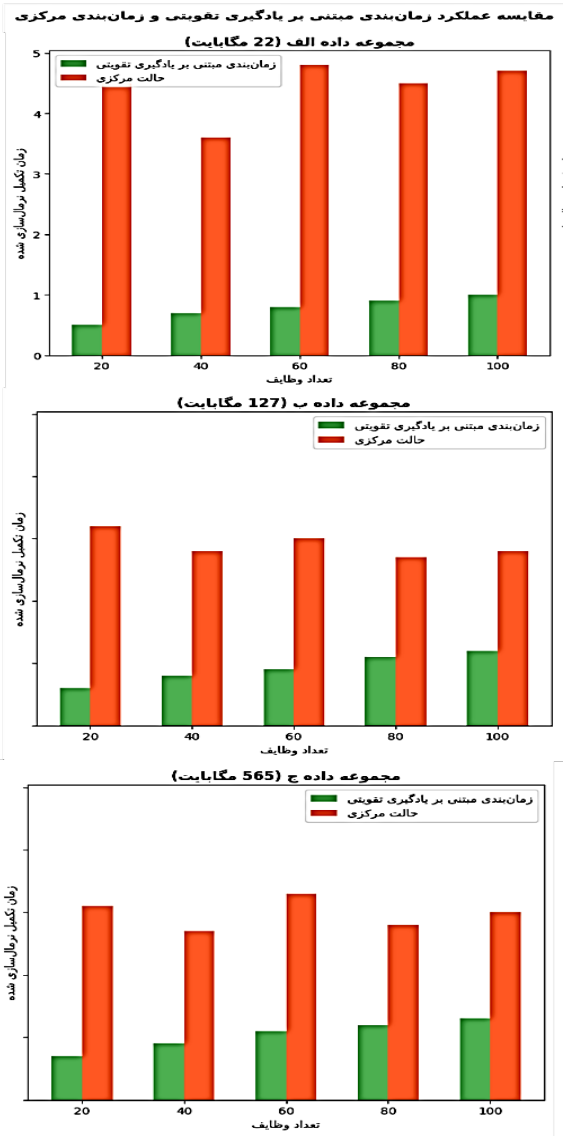
جدول (۳): مشخصات منابع فیزیکی سیستم‌های ابر-مه توزیع شده در مناطق جغرافیایی مختلف

مرکز	حافظه (GB)	ذخیره‌سازی (TB)	تعداد هسته‌ها	پهنای باند (Mbps)	سرعت پردازنده (MIPS)	تعداد ماشین‌های مجازی
مه ۱	۱۲	۱	۱۴	۱۰۰	۵۱۲۰	۲
مه ۲	۱۶	۱	۱۶	۱۰۰	۵۱۲۰	۴
مه ۳	۱۸	۱	۱۸	۱۰۰	۲۸۰۰	۶
ابر	۲۰۰	۱۰۰	۲۰۰	۱۰۰۰۰	۴۴۸۸۰	۱۳

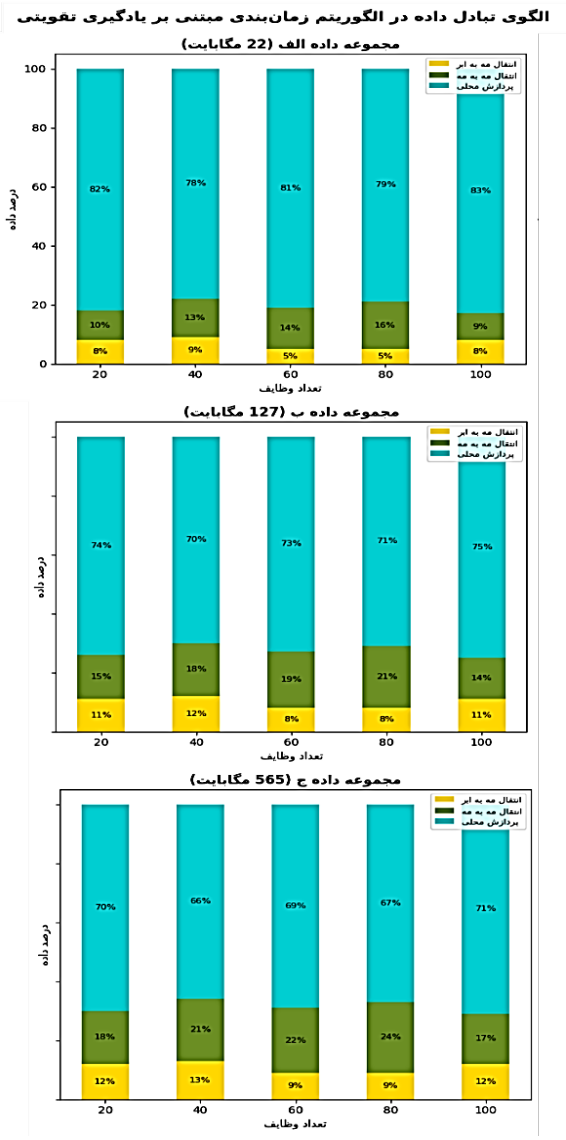


شکل (۵): مقایسه عملکرد زمانبندی الگوریتم یادگیری تقویتی در یک سیستم ابر-مه برای به حداقل رساندن زمان انجام وظایف بر روی سه مجموعه داده مختلف با حجم های ۲۲، ۱۲۹و ۵۶۵ مگابایت

[1] Communication Cost
[2] Data Exchange Pattern
[3] Energy Consumption
[4] Runtime Analysis
[5] Scalability
[6] Lp-Locality
[7] Lp-Migration



شکل (۶): مقایسه عملکرد زمانبندی الگوریتم یادگیری تقویتی با حالت مرکزی برای به حداقل سراندن زمان انجام وظایف بر روی سه مجموعه داده مختلف با حجم های ۲۲، ۱۲۷ و ۵۶۵ مگابایت



شکل (۷): نمونه تبادل داده در الگوریتم یادگیری تقویتی برای ۳ مجموعه داده با حجم های ۲۲، ۱۲۷ و ۵۶۵ مگابایت

در شکل (۶) به مقایسه زمان انجام وظایف برای الگوریتم‌های یادگیری تقویتی و حالت مرکزی پرداخته شده.

الگوی انتقال داده

- هدف: بررسی حجم داده‌های منتقل‌شده بین مراکز.
- نتیجه: ۸۵ ٪ از داده‌ها به صورت محلی پردازش شدند و تنها ۱۵٪ از داده‌ها بین مراکز انتقال یافت.
- تحلیل: این نتیجه نشان می‌دهد که روش یادگیری تقویتی از استراتژی‌های انتقال هوشمندانه استفاده می‌کند و انتقال غیرضروری داده را کاهش می‌دهد.

در شکل (۷) نمونه تبادل داده برای الگوریتم یادگیری تقویتی بررسی شده است.

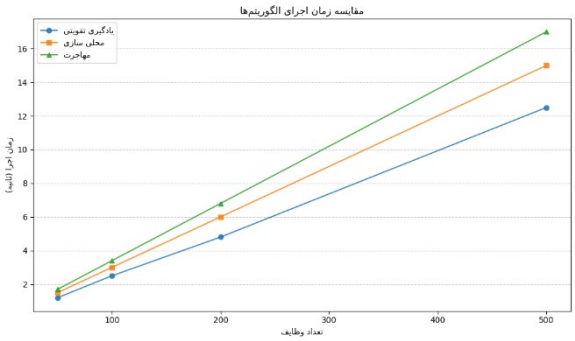
زمان اجرای الگوریتم:

هدف: مقایسه مدت زمان اجرای روش‌های یادگیری تقویتی، مهاجرت و محلی سازی

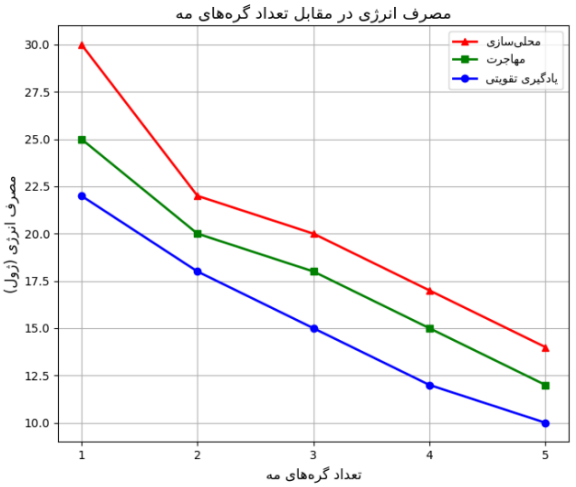
نتیجه: زمان اجرای یادگیری تقویتی به‌طور میانگین ۳ برابر سریع‌تر از در روش ذکر شده است.

تحلیل: الگوریتم یادگیری تقویتی به جای حل مسائل برنامه‌ریزی خطی[1]، از یک سیاست یادگیری استفاده می‌کند که نیاز به زمان محاسباتی کمتری دارد (شکل ۸).

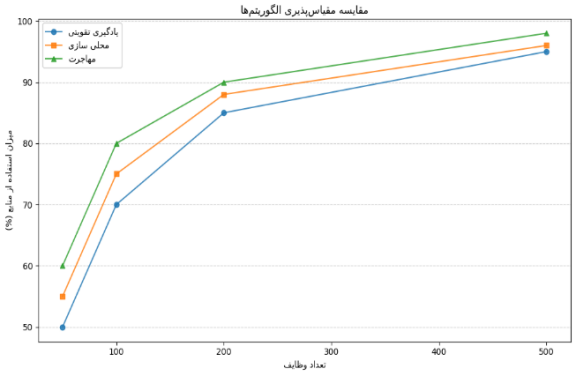
[1] Linear Programming



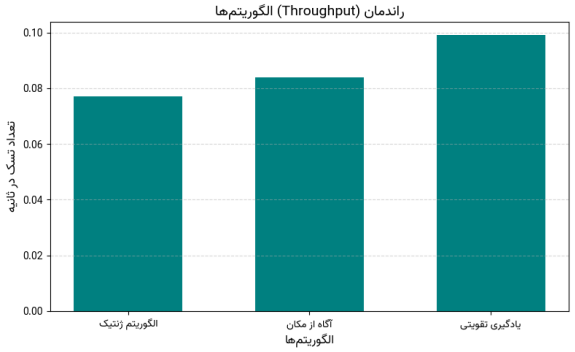
شکل (۸): زمان اجرای الگوریتم با تعداد وظایف متفاوت



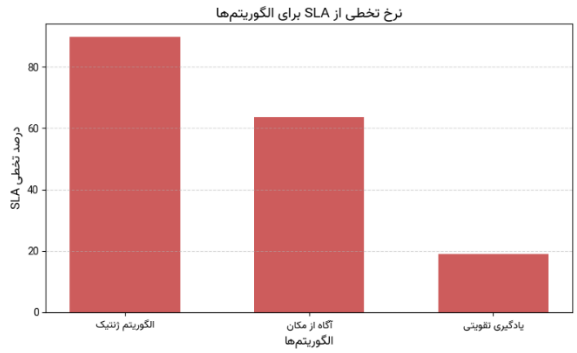
شکل (۹): مصرف انرژی با تعداد گره های متفاوت



شکل (۱۰): مقیاس‌پذیری بر حسب درصد استفاده از منابع



شکل (۱۱): راندمان الگوریتم‌ها بر اساس تعداد وظیفه بر ثانیه



شکل (۱۲): نرخ تخطی برای الگوریتم‌ها

مصرف انرژی

- **هدف:** بررسی میزان انرژی مصرفی در کل سیستم.
- **نتیجه:** روش یادگیری تقویتی مصرف انرژی را ۳۰ درصد نسبت به حالت مرکزی کاهش داده است.
- **تحلیل:** به دلیل پردازش محلی وظایف، نیازی به انتقال داده‌های سنگین به ابر نیست و این موضوع باعث صرفه‌جویی در مصرف انرژی می‌شود.

در شکل (۹) میزان مصرف انرژی سه الگوریتم با تعداد گره‌های مختلف بررسی شده است.

مقیاس‌پذیری

- **هدف:** بررسی توانایی سیستم در مواجهه با بارهای کاری متغیر.
- **نتیجه:** روش یادگیری تقویتی می‌تواند به‌طور پویا منابع جدید را اضافه یا حذف کند.
- **تحلیل:** با افزایش تعداد وظایف از ۵۰ به ۵۰۰، روش یادگیری تقویتی بدون کاهش قابل توجه در عملکرد، وظایف را مدیریت کرد (شکل ۱۰).

ارزیابی راندمان و تخطی از SLA

ارزیابی عملکرد راهکار پیشنهادی بر پایه‌ی یک مجموعه‌داده‌ی شبیه‌سازی‌شده انجام شد. دو معیار کلیدی شامل راندمان سیستم (Throughput) و نرخ تخطی از توافق‌نامه‌ی سطح خدمات (SLA Violation Rate) بررسی گردید. بر اساس تعریف راندمان به‌صورت «تعداد وظایف بر واحد زمان، با توجه به میانگین‌های مشاهده‌شده برای زمان اجرا (ثانیه/وظیفه)، راندمان تقریبی هر الگوریتم به‌صورت معکوس زمان اجرا گزارش می‌شود:

- RL ≈ 0.099 وظیفه/ثانیه (میانگین زمان اجرا ۱۰.۱۱۰ ثانیه)،
- Location-Aware ≈ 0.084 وظیفه/ثانیه (میانگین ۱۱.۸۵۱)،
- GA ≈ 0.077 وظیفه/ثانیه (میانگین ۱۲.۹۶۵).

در شکل (۱۱-۱۲) نتایج را به‌صورت نمودارهای مقایسه‌ای نشان می‌دهند. لینک دیتاست در ادامه آورده شده است.

لینک دسترسی به مجموعه داده

لینک دسترسی به پیاده سازی نتایج آماری

تحلیل آماری

برای زمان اجرای وظایف از آزمون t مستقل با تصحیح Welch (دوطرفه) استفاده شد تا نسبت به ناهمگنی واریانس‌ها مقاوم باشد. سطح معناداری $\alpha=0.05$ در نظر گرفته شد. دو مقایسه از پیش تعریف شد: RL در برابر GA و RL در برابر Location-Aware. برای شفافیت، در هر مقایسه فقط t-value و p-value گزارش می‌شود و تصمیم معناداری بر مبنای p-value اتخاذ می‌گردد. علاوه بر مقدار p، اختلاف میانگین، فاصله‌ی اطمینان ۹۵٪ و اثر اندازه گزارش شد.

یادگیری تقویتی در برابر الگوریتم ژنتیک:

$$\Delta\bar{x} = -2.855("s"), \quad t(1992.7) = -32.043,$$
$$p = 4.553 \times 10\hat{}(-182), \quad 95\%\, CI = [-3.030, -2.681],$$
$$g_"Hedges" = -1.433$$

یادگیری تقویتی در برابر آگاه از مکان

$$\Delta\bar{x} = -1.741("s"), \quad t(1991.1) = -19.456,$$
$$p = 2.497 \times 10\hat{}(-77), \quad 95\%\, CI = [-1.916, -1.565],$$
$$g_"Hedges" = -0.870$$

برای نرخ تخطی از SLA از آزمایش دونسبت استفاده شد و نرخ‌هایی که مشاهده شد به این شکل می‌باشد:

$$(p_"RL")\hat{} = 0.189 \sim (189\backslash/1000),$$
$$(p_"GA")\hat{} = 0.897 \sim (897\backslash/1000),$$
$$(p_"LA")\hat{} = 0.636 \sim (636\backslash/1000)$$

"ژنتیک الگوریتم برابر در تقویتی یادگیری": $z = -31.781, \quad p < 10\hat{}(-200)$

"مکان − آگاه برابر در تقویتی یادگیری": $z = -20.304, \quad p < 0.001$

بنابراین، کاهش نرخ تخطی در روش RL از نظر آماری معنادار است.

۵- نتیجه

در این پژوهش، یک رویکرد نوآورانه مبتنی بر یادگیری تقویتی برای زمان‌بندی وظایف در سیستم‌های توزیع‌شده **ابر–مه** ارائه شد. این رویکرد بر بهینه‌سازی تخصیص وظایف و استفاده هوشمندانه از منابع محاسباتی در لایه‌های مه و ابر متمرکز است. اهداف اصلی این روش شامل کاهش زمان کلی اجرای وظایف، بهبود عملکرد شبکه، کاهش مصرف انرژی و افزایش مقیاس‌پذیری سیستم بود.

کاهش زمان اجرای کل:

- روش یادگیری تقویتی توانست مدت زمان اجرای کل وظایف را به‌طور میانگین ۲۰ درصد کاهش دهد.
- این کاهش ناشی از توانایی این روش در تخصیص هوشمند وظایف و کاهش انتقال داده‌ها بین مراکز مه و ابر بود.
- در مقایسه با الگوریتمDLJSF[2] عملکرد بهتری در شرایط پویا ارائه کرده است.

کاهش مصرف انرژی:

- مصرف انرژی سیستم با استفاده از این روش 30٪ کمتر از روش‌های سنتی بود.
- به دلیل کاهش انتقال داده‌ها و استفاده از منابع محلی مه به جای انتقال داده‌ها به ابر، مصرف انرژی بهینه شد.

مقیاس‌پذیری

- روش یادگیری تقویتی توانست افزایش بار کاری (از ۵۰ به ۵۰۰ وظیفه) را بدون کاهش عملکرد مدیریت کند.
- سیستم به‌طور خودکار منابع جدید را اضافه یا حذف کرد و توانست تغییرات بار را در زمان واقعی مدیریت کند.

نرخ تخطی

- روش یادگیری تقویتی توانست نرخ تخطی از SLA را به‌طور چشمگیری کاهش دهد.
- این کاهش بیانگر پایداری و قابلیت اطمینان بیشتر سیستم در مواجهه با بارهای کاری متغیر است.
- بررسی‌های آماری نیز نشان دادند که این برتری نسبت به روش‌های دیگر از نظر آماری بسیار معنادار بوده و اتفاقی نیست.

مزایای اصلی روش پیشنهادی

- تطبیق‌پذیری پویا: استفاده از یادگیری تقویتی به سیستم این امکان را داد که با تغییرات محیطی و بار کاری تطبیق یابد.
- پردازش محلی: کاهش انتقال داده‌ها به ابر باعث کاهش هزینه‌ها، تأخیرها و مصرف انرژی شد.
- تعادل بار: تخصیص هوشمند وظایف به منابع کم‌بارتر به بهبود عملکرد کلی سیستم کمک کرد.
- کاهش هزینه‌ها: با کاهش مصرف پهنای باند و انتقال داده‌ها، هزینه‌های عملیاتی سیستم کاهش یافت.

مراجع

[1] E. N. Zijiang Hao, "Challenges and Software Architecture for Fog Computing," *IEEE Internet Computing,* vol. 21, no. 2, pp. 44-53, 2017.

[2] M. R. Albert Jonathan, "Nebula: Distributed Edge Cloud for Data-Intensive Computing," *IEEE Transactions on Parallel and Distributed Systems,* vol. 28, no. 11, p. 3229–3244, 2017.

[3] J. S. Rishika Mehta, "Task Scheduling for Improved Response Time of Latency Sensitive Applications in Fog Integrated Cloud Environment," *Multimedia Tools and Applications,* vol. 82, p. 32305–32328, 2023.

[4] N. L. S. d. F. Judy C. Guevara, "Task Scheduling in Cloud-Fog Computing Systems," *Peer-to-Peer Networking and Applications,* vol. 14, p. 962–977, 2021.

[5] R. X. Yao Wang, "Wireless Multiferroic Memristor with Coupled Giant Impedance and Artificial Synapse Application," *Advanced Electronic Materials,* vol. 8, no. 7, 2022.

[6] X. Yu, "Location-aware job scheduling for IoT systems using cloud and fog," *Alexandria Engineering Journal,* vol. 110, pp. 346-362, 2024.

[7] D. L. Gang Sun, "Live Migration for Multiple Correlated Virtual Machines in Cloud-Based Data Centers," *IEEE Transactions on Services Computing,* vol. 11, no. 2, p. 279–291, 2018.

[8] Z. X. Gang Sun, "Dynamic Network Function Provisioning to Enable Network in Box for Industrial Applications," *IEEE Transactions on Industrial Informatics,* vol. 17, no. 10, p. 7155–7164, 2021.

[9] Z. W. Gang Sun, "Profit Maximization of Independent Task Offloading in MEC-Enabled 5G Internet of Vehicles," *IEEE Transactions on Intelligent Transportation Systems,* vol. 25, no. 11, p. 16449–16461, 2024.

[10] Y. L. Yunfei Li, "Variational Bayesian Learning-Based Localization and Channel Reconstruction in RIS-Aided Systems," *IEEE Transactions on Wireless Communications,* vol. 23, no. 9, p. 11309–11319, 2024.

[11] Z. Z. Yang Yang, "Design of a Simultaneous Information and Power Transfer System Based on a Modulating Feature of Magnetron," *IEEE Transactions on Microwave Theory and Techniques,* vol. 71, no. 2, p. 907–915, 2023.

[12] M. T. Marzieh Khosravi, "Diagnosis and Classification of Disturbances in the Power Distribution Network by Phasor Measurement Unit Based on Fuzzy Intelligent System," *The Journal of Engineering,* 2024.

[13] P. P. Xiuwen Fu, "Tolerance Analysis of Cyber-Manufacturing Systems to Cascading Failures," *ACM Transactions on Internet Technology,* vol. 23, no. 4, 2023.

[14] Z. Q. Song Zha, "A Gain-Reconfigurable Reflector Antenna With Surface-Mounted Field-Induced Artificial Magnetic Conductor for Adaptive HIRF Prevention," *IEEE Transactions on Antennas and Propagation,* vol. 72, no. 9, 2024.

[15] S. P. S. Thiruchadai Pandeeswari, "Resource-aware fog service placement using deferred acceptance in edge computing," *Journal of Engineering Research,* 2024.

[16] E. Khezri, "DLJSF: Data-Locality Aware Job Scheduling IoT tasks in fog-cloud computing environments," *Results in Engineering,* vol. 21, 2024.

[17] D. Alsadie, "Advancements in heuristic task scheduling for IoT applications in fog-cloud computing: challenges and prospects," *PeerJ Computer Science,* no. e2128, p. 10, 2024.

[18] A. Umer, "Multi-Objective Task-Aware Offloading and Scheduling Framework for Internet of Things Logistics," *Sensors,* no. 2381, p. 24, 2024.

[19] Salimi, "A Greedy Randomized Adaptive Search Procedure for Scheduling IoT Tasks in Virtualized Fog–Cloud Computing," *Transactions on Emerging Telecommunications Technologies,* no. 5, p. 35, 2024.

[20] F. R. Shahidani, "Task scheduling in edge-fog-cloud architecture: a multi-objective load balancing approach using reinforcement learning algorithm," *Computing,* vol. 105, p. 1337–1359, 2023.