Minimization of latency and energy consumption in cloud-fog hybrid environments based on the schedule of requests in the Internet of Things using the plant defense optimization algorithm

Mustafa Rezaeizadeh Roukerd[1], Mehdi Jafari Shahbazadeh[3], Mahdiyeh Eslami[2]*

UNCORRECTED PROOF

PAPER INFO

Paper history:

Keywords:
Cloud-fog hybrid, plant defenseoptimization algorithm , Internet of Things (IoT), Latency Minimization,
Fog computing

ABSTRACT

Providing services for real-time, latency-sensitive IoT applications is a major challenge. In these applications, there is a possibility of high latency due to the interaction between IoT devices and the physical environment. Fog environments can significantly reduce the latency of services because they move resources to the nearest edge of the network. However, fog nodes are not able to provide all the required resources at a large scale due to energy and processing power limitations. For this reason, the problem of optimizing service requests and energy consumption is raised as a major challenge. The nature of this problem is NP-hard, and therefore, exact optimization solutions are insufficient and impractical for large-scale problems. In this regard, a new approach based on the plant defense algorithm (Plant Defense Optimization Algorithm) is proposed as a novel solution to this challenge. This algorithm is inspired by the genetic defense behavior of plants. In the proposed algorithm, genes are first classified using the K-means clustering method. Then, the plant defense optimization algorithm operators are applied specifically to each solution. These operators are designed to avoid local optimality. In this method, the cost function is calculated as a combination of two factors: time delay and energy consumption. The plant defense optimization algorithm is tested in a simulated environment in which environmental dynamics and changes are carefully considered. Finally, the performance of the plant defense optimization algorithm is evaluated and compared with other methods. The experimental results show that the overall delay in the proposed approach is improved between 23.84% and 48.51% compared to other algorithms.

© 2024 Published by Semnan University Press.
This is an open access article under the CC-BY 4.0 license.(https://creativecommons.org/licenses/by/4.0/)

1 . Department of Electrical and Electronics Engineering, Islamic Azad university of Kerman, Kerman..
2 . Member of the academic staff and director of the electrical engineering and medical engineering department of the Islamic Azad University, Kerman. M_j_shahbazi@yahoo.com
3 . Member of the Faculty of Electrical Engineering Department of Islamic Azad University, Kerman.

کمینه سازی تاخیر درخواست ها در محیط های ترکیبی ابری و مهمی در اینترنت اشیا با استفاده از الگوریتم بهینه سازی دفاع گیاهان

مصطفی رضایی زاده رویکرد[۱] ، مهدی جعفری شهباز زاده*[۲]، مهدیه اسلامی[۳]

اطلاعات مقاله

دریافت مقاله: ۱۳۹۵/۰۳/۲۰
بازنگری مقاله: ۱۳۹۵/۰۶/۲۰
پذیرش مقاله: ۱۳۹۵/۰۷/۱۵

واژگان کلیدی:
محیط ابر-مه، الگوریتم بهینه‌سازی دفاع گیاهی، اینترنت اشیا، کمینه سازی تأخیر، محیط مه

چکیده

ارائه خدمات برای برنامه‌های بلادرنگ اینترنت اشیا که به تأخیر حساس هستند، چالشی اساسی به شمار می‌رود. در این برنامه‌ها، به دلیل تعامل میان دستگاه‌های اینترنت اشیا و محیط فیزیکی، احتمال بروز تأخیرهای زیاد وجود دارد. محیط مه می‌توانند تأخیر سرویس‌ها را به طور قابل توجهی کاهش دهند؛ زیرا منابع را به نزدیک‌ترین لبه شبکه منتقل می‌کنند. با این حال، نودهای مه به دلیل محدودیت‌های انرژی و توان پردازشی، قادر به ارائه تمام منابع مورد نیاز در مقیاس بزرگ نیستند. به همین دلیل، مسئله بهینه‌سازی درخواست سرویس و مصرف انرژی به عنوان یک چالش اصلی مطرح می‌شود. ماهیت این مسئله از نوع NP-hard است و به همین دلیل، راه‌حل‌های بهینه‌سازی دقیق برای مسائل با اندازه‌های بزرگ ناکافی و غیرعملی به شمار می‌روند. در این راستا، یک رویکرد جدید مبتنی بر الگوریتم دفاع گیاهان (الگوریتم بهینه‌سازی دفاع گیاهان) به عنوان راه‌حلی نوین برای این چالش پیشنهاد می‌شود. این الگوریتم از رفتار دفاعی ژنتیکی گیاهان الهام گرفته است. در الگوریتم پیشنهادی، ابتدا ژن‌ها با استفاده از روش خوشه‌بندی K-means دسته‌بندی می‌شوند. سپس، عملگرهای الگوریتم بهینه‌سازی دفاع گیاهان به طور اختصاصی روی هر جواب اعمال می‌گردند. این عملگرها به گونه‌ای طراحی شده‌اند که از بهینگی محلی جلوگیری کنند. در این روش، تابع هزینه به‌طور ترکیبی از دو عامل تأخیر زمانی و مصرف انرژی محاسبه می‌شود. الگوریتم بهینه‌سازی دفاع گیاهان در یک محیط شبیه‌سازی شده آزمایش می‌شود که در آن، دینامیک و تغییرات محیطی به دقت لحاظ می‌شود. در نهایت، عملکرد الگوریتم الگوریتم بهینه‌سازی دفاع گیاهان ارزیابی و با سایر روش‌ها مقایسه می‌شود. نتایج آزمایش‌ها نشان می‌دهند که تأخیر کلی در رویکرد پیشنهادی نسبت به سایر الگوریتم‌ها بین ۲۳.۸۴٪ تا ۴۸.۵۱٪ بهبود یافته است.

DOI:

© 2024 Published by Semnan University Press.
This is an open access article under the CC-BY 4.0 license.(https://creativecommons.org/licenses/by/4.0/)

UNCORRECTED PROOF

۱ . دانشجوی دکتری برق ، دانشگاه آزاد اسلامی واحد کرمان.
۲ . عضو هیات علمی و مدیر گروه مهندسی برق و مهندسی پزشکی دانشگاه آزاد اسلامی واحد کرمان. M_j_shahbazi@yahoo.com
۳ . عضو هیات علمی گروه مهندسی برق دانشگاه آزاد اسلامی واحد کرمان.

۱- مقدمه

پیشرفت و گسترش سیستم‌های اطلاعاتی فناوری اطلاعات و ارتباطات تأثیر گسترده‌ای بر جامعه گذاشته است. این پیشرفت در تکنولوژی مزایا و فرصت‌های زیادی را برای فعالیت‌های علمی، کسب‌وکار و کاربر نهایی فراهم کرده است. اما از طرفی سبب ایجاد داده‌های زیادی از انواع مختلف شده است که یکی از بزرگ‌ترین چالش‌هایی است که معماری‌های اینترنت اشیا و ابری قادر به مقابله با آن نیستند. عوامل دیگر همچون مقیاس بزرگ، تنوع و سرعت کلان داده‌ها، ناهمگونی دستگاه‌های اینترنت اشیا، پروتکل‌های اتصال متفاوت و عدم استانداردهای ارتباطی و تأخیر بالا از جمله ضعف‌هایی هستند که معماری‌های مبتنی بر اینترنت اشیا و ابر با آن مواجه هستند. برای غلبه بر این مشکلات، سیسکو برای اولین بار یک اصطلاح جدید را در سال ۲۰۱۲ با نام محیط مه معرفی کرد. محیط مه یک معماری محاسباتی توزیع‌شده است که در لایه میانی بین مراکز داده ابری و دستگاه‌های اینترنت اشیا قرار دارد، به طوری که خدمات مبتنی بر ابری را می‌توان به دستگاه‌های اینترنت اشیا نزدیک‌تر کرد [۱]. محیط مه با قرارگیری مابین ابر و گره‌های حسگر، موجب جمع‌آوری، پردازش و ذخیره‌سازی داده‌ها به صورت محلی می‌شود و لذا فقط در مواقع ضروری داده‌ها به ابر ارسال می‌گردد. محیط مه با وجود تعداد بسیار زیادی گره حسگر، انرژی مصرفی کمتری نسبت به سیستم محاسبات ابری متمرکز دارد. این‌گونه شبکه‌ها دارای معماری سلسله‌مراتبی از گره‌های حسگر، ابزارهای لبه و ابر هستند. حسگرها در پایین‌ترین سطح از معماری شبکه، داده‌های جمع‌آوری‌شده را به سطح بالاتر ارسال می‌کنند. دستورالعمل‌ها در مه شامل ارسال داده‌ها، پردازش در ابزارهای مه، تقسیم‌کردن یک برنامه به چندین ماژول و تخصیص منابع به همه آن‌ها برای اجرا شدن است. برنامه‌ها برای جمع‌آوری و ذخیره‌سازی داده‌ها از مراکز داده کوچک استفاده می‌کنند تا در آینده مورد پردازش و تحلیل قرار گیرند. محیط مه نسبت به معماری کلاسیک ابری، مزایای زیادی دارد؛ از جمله بهبود کارایی، کاهش مصرف انرژی، کاهش بار ترافیک و افزایش تحرک و کیفیت خدمات سیستم [۲-۴]. برای پشتیبانی از خدمات مه، مدیریت و عملیات جدیدی مانند زمان‌بندی کارها معرفی شده است. این توابع جدید مه با مجموعه‌ای از سرورهای مه که در لبه شبکه، در مکان‌های انتخاب‌شده مانند دفاتر محلی، مراکز خرید یا ایستگاه‌های خدمات قرار می‌گیرند، ارائه می‌شوند. زمان‌بندی نیاز به یکپارچه‌سازی انواع متفاوتی از اطلاعات پردازش مدل‌ها، تعریف ارتباطات بین فعالیت‌ها و منابع، و تعریف اهداف و معیارهای ارزیابی دارد که همه این موارد به هم وابسته هستند. الگوریتم‌هایی که برای حل دقیق این مسائل ارائه شده‌اند، با افزایش اندازه یا محدودیت‌های مسئله، به زمان زیادی نیاز دارند. در سال‌های اخیر، یکی از مهم‌ترین و امیدبخش‌ترین تحقیقات، روش‌های فراابتکاری برگرفته از طبیعت بوده است؛ این روش‌ها شباهت‌هایی با سیستم‌های اجتماعی یا طبیعی دارند. کاربرد این روش‌ها برگرفته از روش‌های ابتکاری پیوسته است که در حل مسائل ترکیبی نتایج بسیار خوبی داشته‌اند. در اینجا به برخی از آن‌ها اشاره می‌شود.

Bitam و همکاران در سال ۱۵ یک الگوریتم مبتنی بر زندگی زنبور عسل جهت زمان‌بندی کارها در محیط مه پیشنهاد دادند. در این مقاله، زمان‌بندی وظایف با استفاده از الگوریتم هیورستیک ترکیبی در محیط مه ارائه شده است. هدف از زمان‌بندی کارها در بین گره‌های مه، به حداقل رساندن تأخیر و بهینه‌سازی مصرف انرژی است Oma .و همکاران در [۷] به مدل‌های تحمل‌پذیر خطای محیط محاسباتی مه در اینترنت اشیا پرداخته‌اند. آن‌ها از دو روش، یکی بدون تکثیر و دیگری با تکثیر استفاده کردند. مدل پیشنهادی از نظر انرژی مصرفی و زمان اجرا مورد بررسی قرار گرفت. در [۸] زمان‌بندی وظایف و تخصیص منابع در محیط فاگ محاسباتی بررسی شده است. در این مقاله، یک الگوریتم زمان‌بندی کار برای اطمینان از تکمیل کار و پردازش همزمان کارها در گره مه ایجاد شد. در پایان، یک مکانیزم تخصیص مجدد منابع برای کاهش تأخیر پردازش کار پیشنهاد شد. نتایج نشان داد که الگوریتم تخصیص مجدد منابع به‌طور مؤثر تأخیر اجرای وظایف را کاهش داده و تعداد همزمان اجرای وظایف در گره‌های مه را بهبود می‌بخشد. در [۹] یک الگوریتم مبتنی بر PSO بهبود یافته برای بهینه‌سازی مسئله زمان‌بندی وظایف در محیط محاسباتی ابر مه پیشنهاد شده است. این الگوریتم پیشنهادی در [۱۱] روی مجموعه داده با اندازه‌های

UNCORRECTED PROOF

مختلف آزمایش شد. نتایج تجربی بهبود ۱۵.۱۱ را نشان می‌دهد. در این مقاله، یک الگوریتم جدید با الهام از زندگی زنبور عسل پیشنهاد شد و تابع برازندگی آن ترکیب خطی زمان اجرای CPU و حافظه تخصیص‌یافته است. Wang و همکاران در [۶] زمان‌بندی منابع در محیط مه بر اساس خوشه‌بندی فازی بررسی کردند. ابتدا ویژگی‌های منابع استاندارد و نرمال می‌شوند، سپس خوشه‌بندی فازی با الگوریتم ازدحام ذرات برای تقسیم منابع ترکیب می‌شود. در این صورت مقیاس جستجوی منابع کاهش می‌یابد. نتایج تجربی نشان می‌دهد که روش مقاله [۱۰] می‌تواند رضایت کاربر و کارایی زمان‌بندی منابع را بهبود بخشد. در [۱۱] روشی برای زمان‌بندی وظایف در محیط فاگ برای شهر هوشمند ارائه شد. آن‌ها زمان اتمام کار و هزینه‌های ارتباطی را با استفاده از الگوریتم ژنتیک چندهدفه کمینه کردند. در [۱۲] یک الگوریتم جدید مبتنی بر تولیدمثل مورچگان برای زمان‌بندی کارها در محیط مه ارائه شد. آن‌ها یک رویکرد جدید بهینه‌سازی مبتنی بر تولیدمثل مورچگان به نام AMO با هدف پرداختن به مسئله زمان‌بندی وظایف در محیط مه پیشنهاد کردند. روش پیشنهادی آن‌ها مبتنی بر توزیع بهینه مجموعه‌ای از وظایف در بین تمام گره‌های محیط مه است. در [۱۳] یک تکنیک زمان‌بندی آگاه از کیفیت سرویس برای پردازش جریان داده در محیط مه پیشنهاد شد. در این کار، نویسندگان اجزای جدیدی نظیر مانیتور کاری، مانیتور کیفیت خدمات و یک زمان‌بندی سازگار را معرفی نمودند. مانیتور کاری مسئول دریافت نرخ ورودی و خروجی برای هر مجری تعریف‌شده است که به‌عنوان یک مؤلفه محاسباتی، یک گروه از وظایف را بر روی گره مه انجام می‌دهد. مانیتور کیفیت خدمات برآورد پارامترهای کیفیت خدمات مانند تأخیر (شبکه) را انجام می‌دهد و مسئول دریافت و استفاده از درون گره و اطلاعات بین گره‌ها است. این اطلاعات به برنامه‌ریزی انطباق توزیع‌شده ارسال می‌شود که سیاست برنامه‌ریزی سیستم را پیاده‌سازی می‌کند. این الگوریتم می‌تواند عملکرد سیستم را بهبود دهد، اما با این وجود، توپولوژی‌های پیچیده مه با تعداد زیاد اپراتورها ممکن است باعث بی‌ثباتی شود که می‌تواند در دسترس بودن پردازش جریان داده را کاهش دهد. در [۱۴]، نویسندگان به مسئله متعادل کردن بار در محیط مه پرداختند تا کیفیت تجربه کاربران را ارتقا دهند. نویسندگان فرض می‌کنند که تمام درخواست‌های کیفیت تجربه کاربران مختلف که نیاز به تخلیه محاسباتی دارند، توسط خوشه‌های محاسباتی محلی منابع اجرا می‌شوند. در این مقاله، یک الگوریتم زمان‌بندی کار برای محیط مه پیشنهاد شده است. اولین قاعده این است که منابع محاسباتی محلی در هر سلول کوچک برای کاربران داخلی اختصاص داده شود. هر سلول کوچک درخواست‌های تخلیه کاربر را بر اساس یک پارامتر خاص مانند زمان ورود، محدودیت زمان تأخیر و غیره مرتب می‌کند. این سفارش همچنین قواعد برنامه‌ریزی را برای تخصیص منابع محلی تعریف می‌کند. این رویکرد رضایت بالای کاربران را از لحاظ کاهش تأخیر و یا مصرف انرژی کم بهبود می‌دهد، اما با افزایش ابعاد مسئله، پیچیدگی محاسباتی بالا می‌رود. در [۱۵] به دفاع بهینه گیاه تحت رقابت برای نور و مواد مغذی پرداخته شده است. گیاهان شورپسند با همسایگان خود برای محدود کردن منابعی مانند نور و نیتروژن رقابت می‌کنند که این امر می‌تواند توانایی رقابتی را به قیمت از دست دادن عملکردهای دیگر مانند دفاع شیمیایی در برابر گیاه‌خواران افزایش دهد. آن‌ها یک مدل شبیه‌سازی جدید را توسعه دادند که یک مدل گیاهی ساختاری-کاربردی از رشد گیاه را در یک محیط آب و هوایی سبک با یک مدل انتخاب طبیعی ترکیب می‌کند. نتایج نشان‌دهنده یک تعامل سه‌جانبه بین رقابت گیاه برای نور، رقابت برای نیتروژن و گیاه‌خواری بود که تعادل بهینه از منابع را تعیین می‌کند. در زیست‌شناسی جدید، الگوریتم پیشنهادی بر اساس مکانیزم‌های دفاع گیاهان است. این الگوریتم اساس مدل شکار-شکارچی ارائه‌شده توسط لوتکا و ولترا را در نظر می‌گیرد که به این معنی است که وقتی گیاه حضور یک ارگانیسم مهاجم را تشخیص می‌دهد، یک سری واکنش‌های شیمیایی را تحریک می‌کند که محصولات آن به هوا منتشر می‌شوند تا غارتگر طبیعی ارگانیسم مهاجم را جذب کنند. عملکرد روش پیشنهادی با بهینه‌سازی مجموعه‌ای از توابع ریاضی تأیید شده و نتایج از نظر آماری در مقابل سایر روش‌های بهینه‌سازی فراابتکاری مقایسه شد. در [۱۶] یک الگوریتم فراابتکاری جدید ارائه شده است که یک الگوریتم بهینه‌سازی الهام گرفته از طبیعت است. هر کدام

UNCORRECTED PROOF

از این مقالات و کارهای پیشین به انواع روش‌های الگوریتم بهینه‌سازی زمان‌بندی و کیفیت سرویس در محیط ابری و مه پرداخته‌اند. این مقالات به طور مثال در مورد حشرات و افت‌های بیرونی است و دفاع گیاهان در مقابل عوامل خارجی است، اما در این مقاله قصد داریم یک روش فراابتکاری از الگوریتم بهینه‌سازی دفاع گیاهان با روش هورمونی ارائه دهیم. تاکنون الگوریتم بهینه‌سازی دفاع گیاهان در رابطه با هورمون‌های گیاه در مقابل عوامل بازدارنده رشد ارائه نشده است. علاوه بر این، در مطالعات پیشین نشان داده شده است که بهینه‌سازی تأخیر به همراه کمینه‌سازی مصرف انرژی مورد بررسی قرار نگرفته است. برای حل مسئله کمینه‌سازی تأخیر درخواست، در این مقاله از روش الهام گرفته‌شده از دفاع گیاهان (الگوریتم بهینه‌سازی دفاع گیاهان) استفاده شد. ابتدا یک کاربر ساکن یا متحرک یک درخواست سرویس را به یک گره فاگ که در لبه شبکه واقع شده است ارسال می‌کند. سپس گره فاگ داده‌ها و پارامترهای این درخواست را به عنوان یک کار به گره مدیر که اغلب دور از کاربر قرار دارد ارسال می‌کند. گره مدیر درخواست را به مجموعه‌ای از وظایف تجزیه می‌کند و پس از آن الگوریتم بهینه‌سازی دفاع گیاهان اجرا می‌شود تا بهینه‌ترین زمان‌بندی و تخصیص را بر اساس تابع هدف پیدا کند. از آنجایی که کمینه‌سازی مصرف انرژی در گره‌ها مهم است، در اینجا از ترکیب زمان اجرا و مصرف انرژی استفاده شده است. گره مدیر به هر گره فاگ دسترسی دارد و می‌تواند وظایف را برای هر گره فاگ ارسال کند. هر گره فاگ نتایج را به گره مدیر می‌دهد. مفروضات این مقاله به شرح زیر است:

- گره در این مقاله نشان‌دهنده یک گره مه است.
- هر گره دارای حافظه محدود است.
- هر وظیفه را می‌توان به هر گره اختصاص داد.

مصرف انرژی در هر گره با توجه به مدل مصرف انرژی [۱۷] محاسبه می‌شود. با توجه به اهمیت تعاملات پیچیده میان هورمون‌های گیاهی در پاسخ به حملات پاتوژن‌ها، در این تحقیق به بررسی الگوریتم‌های دفاعی گیاهان و نحوه‌ی بهینه‌سازی این فرآیندها پرداخته شده است. ترکیب هورمون‌های مختلف مانندET ، JA و ABA و تأثیر آن‌ها بر تقویت ایمنی گیاهان، به‌ویژه در برابر عوامل بیماری‌زا، از جمله مباحث کلیدی در این پژوهش است. الگوریتم دفاع گیاهان با استفاده از مدل‌های بهینه‌سازی، هدف خود را بر کاهش تأخیر پردازش درخواست‌ها و مصرف انرژی در محیط‌های محاسباتی فاگ-ابر متمرکز کرده است. در این راستا، مدل‌های پیشنهادی به‌طور مؤثر با استفاده از الگوریتم‌های بهینه‌سازی چندهدفه مانندPMO ، به دنبال دستیابی به بهترین عملکرد در کاهش تأخیر و مصرف انرژی هستند. این تحقیق نشان می‌دهد که با استفاده از روش‌های پیشرفته الگوریتمی، می‌توان به بهبود و تقویت سیستم‌های دفاعی گیاهان در برابر پاتوژن‌ها به‌طور همزمان با بهینه‌سازی منابع محاسباتی دست یافت. در ادامه، مقاله به شرح زیر سازمان‌دهی شده است: در بخش ۲ و ۳ مدل سیستم و مدل زمان‌بندی کار پیشنهادی را توضیح می‌دهیم، سپس جزئیات پیاده‌سازی و نتایج ارزیابی الگوریتم پیشنهادی را در بخش ۴ مورد بحث قرار می‌دهیم. در نهایت، بخش ۵ نتیجه‌گیری و بینش‌هایی را برای تحقیقات آینده ارائه می‌دهد.

۱-۱- سیستم و مدل کاری

جدول ۱ علائم مورد استفاده در تشریح الگوریتم را به همراه تعاریف آن‌ها نشان می‌دهد. رویکرد اصلی روش پیشنهادی به شرح زیر است:

روش پیشنهادی یک روش برای حل مسئله کمینه‌سازی تأخیر درخواست و مصرف انرژی در محیط مه-ابر است. برای این هدف، یک رویکرد بهینه‌سازی جدید الهام‌گرفته از طبیعت به نام الگوریتم دفاع گیاهان (الگوریتم بهینه‌سازی دفاع گیاهان) پیشنهاد شده است. به این ترتیب، زمان تأخیر پاسخ و هزینه انرژی می‌تواند درخواست کاربران را تأمین کند. رویکرد بهینه‌سازی پیشنهادی مبتنی بر الگوریتم بهینه‌سازی دفاع گیاهان را ارزیابی می‌کنیم و کارایی آن را با مقایسه آن با رویکردهای دیگر ارزیابی می‌کنیم.

UNCORRECTED PROOF

جدول ۱ علائم مورد استفاده به همراه تعریف آن‌ها

نشان	تعریف	نشان	تعریف
N	تعداد درخواست یا کارها	T_R	دریافت نتایج
R	درخواست	EC(x)[J]	مصرف انرژی الکتریکی بر حسب ژول
r_k	وظیفه k م از درخواست R	Z	اطلاعات ارسالی به گره Admin
FN	نود مه	PI	توان الکتریکی ماژول ورودی
Admin	نود مدیر	PO	توان الکتریکی ماژول خروجی
q_t	میانگین زمان انتظار در صف	PS[W]	توان الکتریکی بر حسب وات برای ذخیره‌سازی داده
CET_t	زمان پردازش	ES	مصرف انرژی الکتریکی
T_S	زمان ارسال		

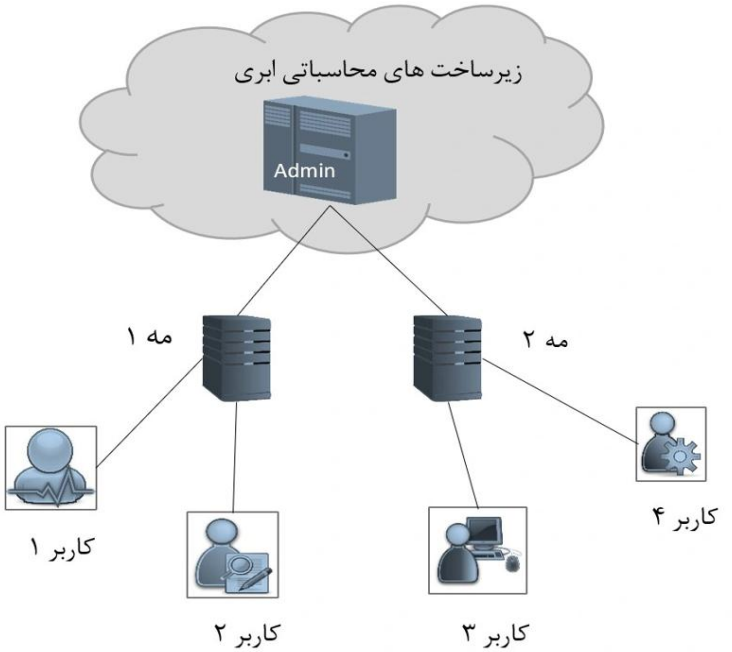
۱-۲- تعریف مسئله

در اینجا، مسئله کمینه‌سازی تأخیر درخواست در محیط مه-ابر عبارت است از تخصیص مجموعه‌ای از وظایف یا کارها به گره‌های فاگ که در لبه شبکه قرار گرفته‌اند، به‌نحوی که نیازهای کاربران برطرف شود. در این مقاله، به‌طور خاص، گره‌های فاگ برای کارهای مختلف درخواست‌شده از سوی کاربران سیار یا ساکن، طوری انتخاب می‌شوند که زمان تأخیر و انرژی مصرفی به حداقل برسد. بنابراین، هدف کمینه‌سازی زمان تأخیر و مصرف انرژی است. در این بخش، مدل سیستم برای زیرساخت محاسبات فاگ را ارائه می‌کنیم.

۱-۳- مدل سیستم

همان‌طور که در شکل ۱ نشان داده شده است، سیستم محاسباتی شامل m گره ثابت مه واقع در ابر یا شبکه است. گره‌های مه در سیستم پیشنهادی دسترسی مستقیم به یکدیگر ندارند. سیستم فرض‌شده در این مقاله شامل یک گره برنامه‌ریزی وظیفه است که همه وظایف را با پارامترها، تعداد درخواست‌ها، اطلاعات شبکه و غیره دریافت می‌کند. ما در اینجا این گره را گره مدیر می‌نامیم. این گره اطلاعات تأخیر و مصرف انرژی در هر گره را می‌داند. کاربران سیستم پیشنهادی می‌توانند ثابت یا متحرک باشند. کاربران ثابت می‌توانند کاربران رایانه شخصی باشند و کاربران متحرک شامل تلفن همراه، پزشکان، پرستاران و ... هستند. هر درخواست مجموعه‌ای از وظایف است که توسط یک کاربر متحرک یا ثابت ارائه می‌شود و در زیرساخت محیط مه انجام می‌شود.

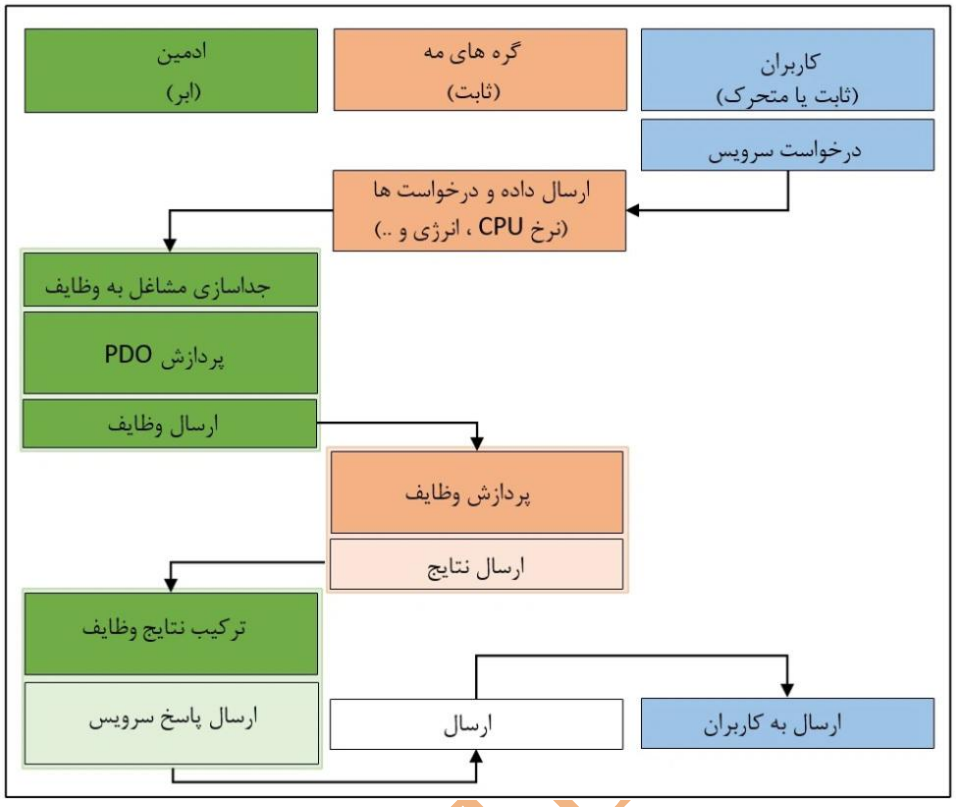
UNCORRECTED PROOF



شکل ۱- مدل پیشنهادی

برای حل مسئله کمینه‌سازی تأخیر درخواست‌ها، یک الگوریتم جدید با الهام از طبیعت به نام الگوریتم بهینه‌سازی دفاع گیاهان پیشنهاد شده است که بر روی گره مدیر اجرا می‌شود تا بهینه‌ترین جواب را برای پردازش توسط گره‌های مه پیدا کند. بنابراین، گره‌های مه می‌توانند برنامه‌ریزی را با زمان پردازش بهینه و رویکرد مصرف انرژی تضمین کنند. مدل سیستم در شکل ۲ با خلاصه‌ای از مراحل اجرای درخواست خدمات نشان داده شده است.

UNCORRECTED PROOF



شکل ۲- **مدل سیستم**

در این مدل، یک کاربر ثابت یا متحرک درخواست خدمات را به یک گره مه که در لبه شبکه این زیرساخت محاسباتی قرار دارد، ارسال می‌کند (مرحله ۱). سپس، گره مه داده‌ها و پارامترهای این درخواست را به‌عنوان یک وظیفه به گره مدیر ارسال می‌کند که اغلب از کاربر دور است (مرحله ۲). گره مدیر درخواست را به مجموعه‌ای از وظایف تقسیم می‌کند (مرحله ۳) و سپس الگوریتم بهینه‌سازی دفاع گیاهان بخش (۳۲) اجرا می‌شود تا برنامه بهینه را پیدا کند (مرحله ۴). گره مدیر به هر گره مه دسترسی دارد و می‌تواند وظایف را به هر گره مه ارسال کند. در مرحله بعد، هر گره مه وظیفه تعیین‌شده خود را دریافت می‌کند (مرحله ۵). این کار در سطح گره انجام می‌شود (مرحله ۶). هر گره مه نتایج خود را (مرحله ۷) به گره مدیر می‌رساند. گره مدیر نتیجه نهایی را بر اساس نتایج بخش‌بندی‌شده (مرحله ۸) و دریافت‌شده از گره‌های مه آماده می‌کند. سپس، نتیجه نهایی به گره مربوطه ارسال می‌شود و در نهایت برای کاربر به‌عنوان پاسخ سرویس ارسال می‌شود (مرحله ۹).

۲- الگوریتم دفاع گیاهان

در طی تکامل، گیاهان سازوکارهای مختلفی را برای مقابله با انواع تنش‌ها، از جمله عفونت‌های میکروبی، به کار می‌گیرند. یکی از این سازوکارها، فعال شدن مسیرهای علامت‌دهی است که منجر به بیان ژن‌های مؤثر در دفاع می‌شود. علامت‌دهی حضور بیمارگر را به‌منظور ایجاد مقاومت مخابره می‌کند. از جمله سازوکارهای دفاعی که در اثر فعال شدن مسیر علامت‌دهی به وجود می‌آید، می‌توان به تقویت دیواره سلولی از طریق توسعه ساختارهای دفاعی خود شامل کوتیکول و تغییر در دیواره سلول، تولید گونه‌های اکسیژن واکنشگر، ایجاد واکنش فوق‌حساسیت، مرگ برنامه‌ریزی‌شده سلول، فعال شدن ژن‌های دفاعی و سنتز متابولیک‌های ثانویه و هورمون‌های دفاعی اشاره کرد. برخلاف هورمون‌های کلاسیک مانند ET ، SA ، ABA و GA ، هورمون‌هایی مانند JA ، اکسین‌ها، BRها و CK ها از لحاظ تاریخی عمدتاً به دلیل نقش آن‌ها در رشد و نمو گیاه مورد مطالعه قرار گرفته‌اند و اخیراً به‌عنوان عوامل تعیین‌کننده کلیدی در نتیجه فعل و

UNCORRECTED PROOF

انفعالات گیاهی و بیماری‌زایی مطرح شده‌اند. در حمله پاتوژن‌ها، گیاهان به سرعت منابع سلولی را از فرایندهای رشد طبیعی به واکنش‌های دفاعی منتقل می‌کنند. در اینجا، برخی از آن‌ها مورد بررسی قرار می‌گیرند.

۲-۱- هورمون SA

هورمون گیاهی فنولی SA به دلیل نقش آن در ترموژنز، گلدهی، سیگنال‌دهی دفاعی و مقاومت اکتسابی عمومی (SAR)مشهور است [۱۸]. با این حال، گونه‌های مختلف گیاهی در سطوح SA درون‌زا و واکنش آن‌ها به این هورمون بسیار متفاوت است. در تنباکو (Nicotiana tabacum) و Arabidopsis، سطوح پایه SA پایین است (حدود ۵۰ نانوگرم در گرم)، اما میزان عفونت پاتوژن می‌تواند دو برابر افزایش یابد. با این حال، برگ‌های سالم برنج بدون تغییرات موضعی یا سیستمی‌ک قابل توجهی در حمله پاتوژن، دارای سطوح پایه بالایی از ۷ میلی‌گرم در گرم وزن تازه هستند. در تنباکو، SAسنتز نشده به سرعت به SA گلیکوزید گلیکوزیله می‌شود، در حالی که در برنج بیشتر SA به شکل اسید آزاد وجود دارد [۱۹][۲۰]. در هسته ۱، AtNPR با فاکتورهای TGA در تعامل است و ژن‌های دفاعی خاصی را فعال می‌کند. SA در برخی گیاهان به مسیرهای وابسته به OSNPR1و OsWRKY45 منشعب می‌شود که نقش مکمل را در دفاع از گیاه ایفا می‌کنند. تقریباً نیمی از ژن‌های پاسخگو به بنزوتیاد یازول و دو سوم ژن‌های تنظیم‌شده با بنزوتیاد یازول، وابسته به OSNPR هستند. در اینجا، ژن‌های مربوط به هورمون SA به سه دسته تقسیم می‌شوند و وزن‌های جدید از رابطه زیر به‌دست می‌آیند.

(۱)

$$\vec{G}_n = \begin{cases} \overrightarrow{G'^{SA}_1} + w_1 \times \vec{r}_1 \times \left(\overrightarrow{G'^{SA}_1} - \overrightarrow{G'^{SA}_2}\right) & \alpha > rand \\ \overrightarrow{G'^{SA}_1} + w_2 \times \vec{r}_2 \times \left(\overrightarrow{G'^{SA}_1} - \overrightarrow{G'^{SA}_2}\right) & \alpha < rand < \beta \\ \overrightarrow{G'^{SA}_1} + w_3 \times \vec{r}_3 \times \left(\overrightarrow{G'^{SA}_1} - \overrightarrow{G'^{SA}_2}\right) & \beta < rand \end{cases}$$

در رابطه ۱، $\beta(\alpha < \beta)$ پارامترهای تعیین‌کننده ژن‌های مربوط به هورمون SA می‌باشند.W وزن و r اعداد تصادفی هستند. JA یک گره مرکزی در شبکه سیگنالینگ دفاع گیاه است. JA و متابولیک‌های آن که در مجموع به عنوان Jasmonates (JA) شناخته می‌شوند، تنظیم‌کننده‌های مهم مشتق‌شده از چربی هستند که نقش اساسی در دفاع و فرآیندهای رشد گیاه دارند. کاربرد JA توسعه بیماری را کاهش می‌دهد و بیوسنتز دست‌نخورده JA برای مقاومت در برابر زخم غلاف ناشی از ویتامین B محلول در آب، در ریبوفلاوین ضروری است. علاوه بر این، عامل بیماری OsWRKY30 با افزایش تجمعJA ، مقاومت سیگنال JA را در برابر نکروتروف‌ها افزایش می‌دهد. با توجه به مدل دفاعی کلاسیک، JAعمدتاً در برابر عوامل بیماری‌زای نکروتروف و حشرات گیاه‌خوار مؤثر است و در برخی موارد، مقاومت بیوتروف با واسطه SA را تضعیف می‌کند[۲۱]. یکی از بهترین نمونه‌های بررسی‌شده در زمینه تداخل سیگنال مربوط به دفاع، تعامل بین مسیرهای پاسخ SA و JAاست [۲۲]. اگرچه تحقیقاتی نشان می‌دهند که مسیرهای SA و JA به صورت متضاد عمل می‌کنند، فعل و انفعالات مثبت نیز گزارش شده است. به‌عنوان مثال، فعال‌سازی بیوسنتز JA در گیاهان جهش‌یافته با کمبود هیدروپراکسید لیاز OSHPL همزمان با افزایش سطح SA همراه است. برای شبیه‌سازی این رویکرد در گیاهان، ابتدا چند ژن از خانواده SA انتخاب می‌کنیم. این ژن‌ها با وزن کمتر از ژن‌های JA با هم ترکیب می‌شوند (۰) و سپس ژن‌های جدید جایگزین ژن‌های قبلی می‌شوند و رابطه ۲ تعریف می‌شود. این عمل بر روی تعداد کمیاب جمعیت SA اعمال می‌شود.

(۲)

$$\vec{G}_n = w_5 \overrightarrow{G'^{JA}_1} + w_4 \times \vec{r}\left(\overrightarrow{G'^{SA}_1} - \overrightarrow{G'^{JA}_1}\right)$$

۲-۲- هورمون دفاعی کلاسیک ET

ET یکی از سه هورمون دفاعی کلاسیک و یکی از اجزای اصلی ترکیب هورمونی است که در حمله پاتوژن آزاد می‌شود. اگرچه استثنائاتی وجود دارد اما به طور گسترده‌ای پذیرفته شده است که ET با JA در تقویت ایمنی در برابر عوامل بیماری‌زای نکروتروف همکاری کند. اندازه‌گیری‌های هورمونی نشان داد که مسیر ET در گیاهان حساس فعال است اما در

UNCORRECTED PROOF

گیاهان مقاوم فعال نیست [۱۹] بنابراین اعضایی از خانواده ET با خانواده JA ترکیب می‌شوند رابطه (۳) و همچنین اعضایی از خانواده JA با خانواده ET ادغام می‌شوند (رابطه ۴).

(۳)

$$\vec{G}_n = \overrightarrow{G'^{JA}} + w \times \vec{r}(\overrightarrow{G'^{SA}} - \overrightarrow{G'^{JA}})$$

(۴)

$$\vec{G}_n^{ET} = \vec{G}^{ET} + w \times \vec{r}(\vec{G}^{ET} - \vec{G}^{JA})$$

۲-۳- هورمون ABA

در مقایسه با JA SA و ET نقش هورمون ABA در ایمنی ذاتی گیاه کمتر شناخته شده است. اگرچه اثرات مثبت و منفی ABA بر مقاومت بیماری گزارش شده است. **اما** ABA عمدتاً به عنوان تنظیم کننده منفی ایمنی عمل می‌کند ۲۳] عفونت معمولاً با برنامه‌ریزی مجدد گسترده ژن‌های ABA و بیوسنتز همراه است و نشان می‌دهد که این عوامل بیماری‌زا مسیر ABA برنج را برای ایجاد بیماری تغییر داده‌اند. از این‌رو در اینجا بر روی تعداد δ درصد از اعضای جامعه جهش به‌صورت زیر رخ می‌دهد.

(۵)

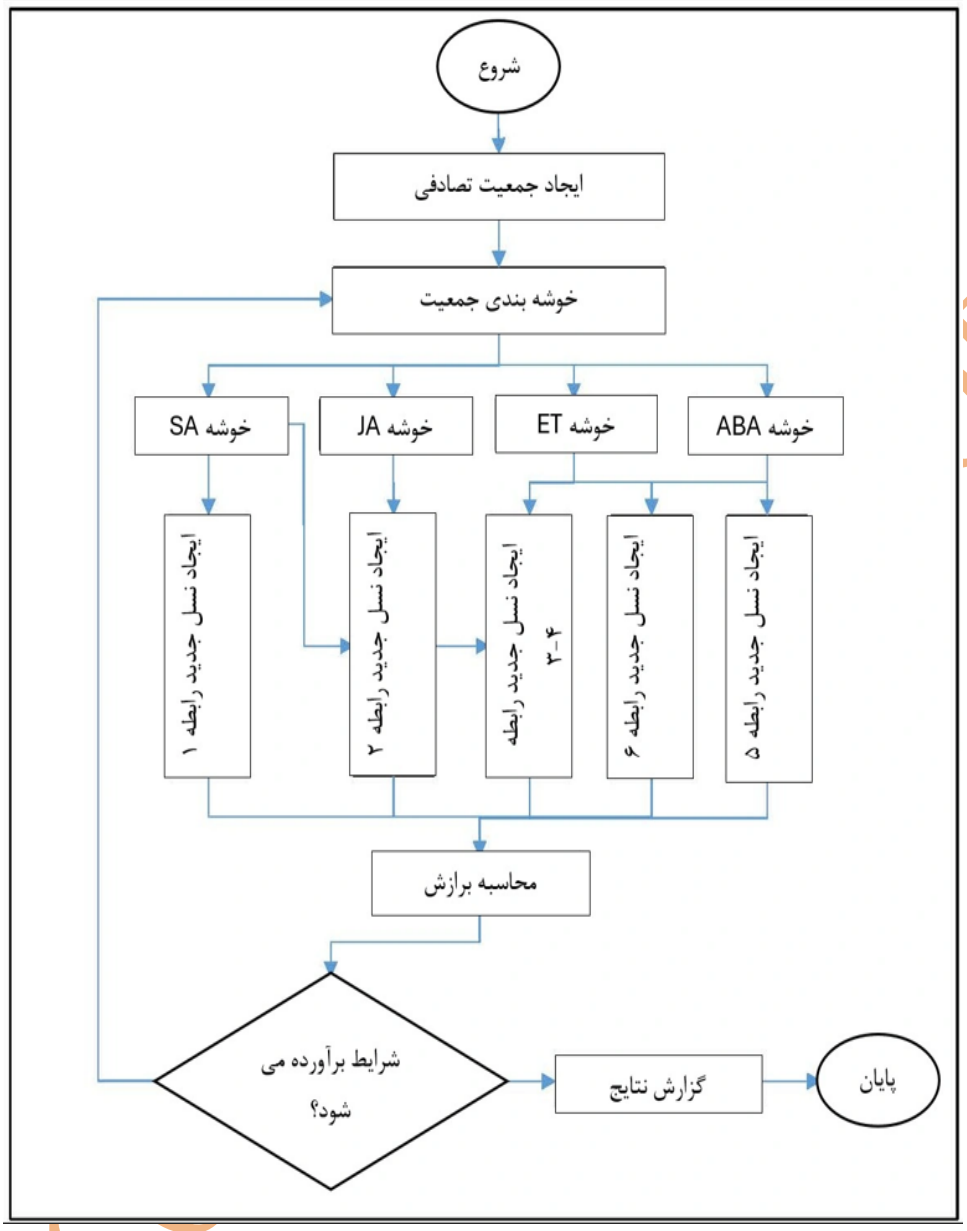
$$\vec{G}_n = +w \times \vec{r}$$

از طرفی خطوطی که برای پروتئین کیناز OsMPK6 که توسط میتوژن فعال شده است توسط ABA ساکت شده است و ET را بیش از حد تولید می‌کند و مقاومت بیشتری در برابر M. oryzae نشان می‌دهد. این رویکرد نشان می‌دهد که تعداد اعضای ET از طریق رابطه زیر افزایش پیدا می‌کنند. جهت تولید ET از رابطه ۶ استفاده می‌شود.

(۶)

$$\vec{G}_n = \vec{G}^{ET} + w_1 \times \vec{r}_1 (\vec{G}^{ET} - \vec{G}^{ET})$$

UNCORRECTED PROOF



شکل ۳- **فلوچارت الگوریتم بهینه‌سازی دفاع گیاهان**

۲-۴- تحلیل پیچیدگی زمانی و مقیاس‌پذیری

الگوریتم پیشنهادی

الگوریتم بهینه‌سازی دفاع گیاهان شامل مراحل خوشه‌بندی اولیه، تکرار فرآیند اختصاص وظایف به گره‌های فاگ (با جستجوی محلی و سراسری)، اعمال عملگرهای تصادفی و ارزیابی برازندگی در هر نسل است. فرض کنیم:

- N تعداد درخواست‌ها یا وظایف
- M تعداد گره‌های فاگ
- G تعداد نسل‌ها (Iterations)
- P اندازه جمعیت (ژنوم‌ها)

در هر نسل، خوشه‌بندی (با روش K-means) دارای مرتبه زمانی $O(N \times K \times I)$ است که K تعداد خوشه‌ها (در اینجا معمولا کوچک مانند ۳ تا ۴) و I تعداد تکرار تا همگرایی (که معمولاً ثابت است).

ارزیابی برازندگی تمام اعضای جمعیت در هر نسل، مرتبه زمانی $O(P \times N \times M)$ دارد، چون هر ژن مشخص می‌کند چگونه وظایف روی گره‌ها توزیع شوند و برازندگی باید برای همه وظایف و سرورها محاسبه گردد.

بنابراین، پیچیدگی کل الگوریتم در حدود:

(۷)
$$O(G \times [P \times N \times M + N \times K \times I]) \approx O(G \times P \times N \times M)$$

است. معمولا مقدار G، P، K و I بصورت ثابت یا روی بازه کوچک انتخاب می‌شود (مثلاً P=50، G=100)، پس پیچیدگی عمدتاً به صورت خطی نسبت به N و M رشد می‌کند (اگر پارامترها مقیاس‌پذیر شوند ممکن است رشد چندجمله‌ای جزئی پیدا کند).

در مقایسه با الگوریتم‌های کلاسیک صف‌بندی مانند WFQ و PSQ که معمولاً مرتبه زمانی $O(N \log N)$ یا کمتر دارند، الگوریتم پیشنهادی هزینه محاسباتی بالاتری دارد (به ویژه به دلیل فرآیند تکراری و خوشه‌بندی).

با این حال، در آزمایش‌های صورت گرفته تا ابعاد ۵۰۰ درخواست و ۱۶ سرور فاگ، الگوریتم در زمان قابل قبولی همگرا می‌شود. انتظار می‌رود با افزایش شدید تعداد وظایف یا سرورها، زمان اجرا به صورت تقریباً خطی (و در حضور همزمان چند نسل و جمعیت، چندجمله‌ای) افزایش یابد که جزو محدودیت‌های ذاتی روش‌های فراابتکاری محسوب می‌شود. راهکارهای مدرن مانند موازی‌سازی ارزیابی برازندگی یا اجرای الگوریتم روی پردازش‌گرهای گرافیکی می‌تواند اجرای مدل را در ابعاد بالا شتاب ببخشد، که بررسی آن‌ها به عنوان بخشی از کارهای آینده قابل توصیه است.

جدول ۲ مقداردهی و تنظیم هایپرپارامترهای الگوریتم پیشنهادی

پارامتر	مقدار انتخاب‌شده	دلیل انتخاب / منبع انتخاب
تعداد سرور فاگ (M)	۱۶	تست در سناریوی متوسط و مقایسه با مقالات مرتبط [۱۷،۱۸]
تعداد درخواست (N)	۱۰۰، ۵۰۰	پوشش شرایط کوچک و بزرگ برای سنجش مقیاس‌پذیری
اندازه جامعه (P)	۵۰	مقدار رایج در الگوریتم‌های فراابتکاری؛ آزمون و خطا برای بالانس دقت و زمان اجرا
تعداد نسل (G)	۱۰۰	مبتنی بر تجربه و حصول همگرایی مناسب پس از این تعداد
تعداد خوشه (K)	۳	براساس ساخت الهامی الگوریتم (۳ خانواده هورمونی)، و پاسخ بهینه‌تر در تست‌ها
وزن تابع هدف (k1, k2)	۰.۵، ۰.۵	برای بالانس یکسان بین تاخیر و انرژی؛ مقادیر دیگر هم تست شد، اما این مقدار نتیجه بهینه داشت
سرعت پردازش سرورها	۵۰۰ بسته/ثانیه (متوسط)	مقدار معمول در شبیه‌سازی‌های مشابه؛ تست شده بین ۵۰ تا ۱۰۰۰ بسته/ثانیه
میانگین اندازه بسته	متغیر (۱۰۰۰–۱۰۰۰۰)	جهت بررسی تأثیر مقیاس و تحمل‌پذیری الگوریتم

مقادیر پارامترهای الگوریتم با ترکیبی از مرور ادبیات موضوع و آزمون‌های اولیه بر اساس مجموعه داده و محیط شبیه‌سازی منتخب شدند. برای اندازه جامعه و تعداد نسل‌ها، چند مقدار مرسوم (P=30, 50, 100 و G=50, 100, 200) مورد تست قرار گرفت که مقدار ۵۰ و ۱۰۰ بین پایداری راه‌حل و سرعت همگرایی، وضعیت مناسبی ایجاد کرد. تنظیم وزن‌ها نیز مبتنی بر تست سناریوهای مختلف (تأکید بر انرژی یا تاخیر) بود و مقدار برابر بالانس بهتری فراهم ساخت. سایر پارامترها مانند تعداد سرورهای فاگ و سرعت، مطابق با دامنه مرسوم مطالعات مشابه و با هدف مقایسه تطبیقی انتخاب شدند.

۳- کمی سازی مسئله

در این بخش ما مسئله کمینه‌سازی تأخیر درخواست را در محیط محاسبات ابر مه تدوین می‌کنیم یک کار به عنوان یک فرایند تعریف می‌شود که مربوط به یک درخواست سرویس

UNCORRECTED PROOF

است که توسط یک کاربر فاگ ایجاد می‌شود که می‌تواند یک کاربر سیار یا ساکن باشد. سپس مجموعه‌ای از n کار به منظور زمانبندی و اجرا در زیر ساخت محاسبات فاگ ثبت می‌شود. این کارها توسط مجموعه کاری زیر تعریف می‌شود:

(۸)

$$Requests = \{r_1, r_2, \dots, r_n\}$$

هر درخواست R در میان n درخواست به‌طوری که $1 \leq i \leq n$ می‌تواند به r وظیفه تقسیم شود. هر درخواست از k وظیفه تشکیل شده است بطوریکه $1 \leq k \leq r$ ، همچنین هر وظیفه می‌تواند جهت پردازش به بود فاگ j از m نود فاگ موجود تخصیص داده شود بطوریکه $(1 \leq j \leq m)$ برای اندیس گذاری از فرمت زیر استفاده می‌شود:

(۹)

$$Request_{R.Index} Task^{Fog.Indesx}_{Task.Index}$$

در زیر نشان داده شده است که درخواست R_i از r وظیفه تشکیل شده است و هر وظیفه به یک نود فاگ جهت پردازش ارسال شده است.

(۱۰)

$$Reques_i Tasks = \{R_i Task_1^a, R_i Task_2^b, \dots, R_i Task_k^j, \dots, R_i Task_r^m\}$$

به عنوان مثال داریم

(۱۱)

$$Reques_i Tasks = \{R_1 Task_1^4, R_1 Task_2^6, \dots, R_1 Task_3^9\}$$

عبارت بالا تعداد وظایف تشکیل دهنده کار اول را نشان می دهد همچنین بیان می‌کند اولین وظیفه $(J_1 Task_1^4)$ در فاگ نود ۴ اجرا می‌شود دومین وظیفه $(J_1 Task_2^6)$ در فاگ نود ۶ اجرا می‌شود و سومین وظیفه $(J_1 Task_3^9)$در فاگ نود ۹ اجرا می شود. در نتیجه هر فاگ نود FN_j می‌تواند مجموعه‌هایی از وظایف گسسته را که از هر کار حاصل می‌شود پردازش کند. نمایش وظایف تخصیص داده شده به فاگ نود FN_j بصورت زیر است.

(۱۲)

$$FN_j Tasks = \{R_a Task_x^j, R_b Task_y^j, \dots, R_i Task_k^j, \dots, R_i Task_r^j\}$$

با کنار هم قرار دادن مجموعه وظایف بر روی هر فاگ بود مجموعه کامل درخواست به دست می‌آید. به عنوان مثال پس از زمانبندی وظایف زیر به گره فاگ FN_j اختصاص داده می‌شود:

(۱۳)

$$FN_j Tasks = \{R_2 Task_3^j, R_6 Task_1^j, R_7 Task_4^j\}$$

بر اساس رابطه بالا نود FN_j، وظیفه سوم از درخواست دو وظیفه اول از درخواست ششم و وظیفه چهارم از درخواست هفتم را پردازش می‌کند. در اینجا فرض می‌شود که تأخیر در پردازش درخواست‌ها می‌تواند سه دلیل داشته باشد: صف، زمان پردازش و زمان ارسال و دریافت نتایج.

۳-۱- صف (q_t)

بعد از ارسال درخواست برای یک پردازنده مدت زمانی که طول می‌کشد تا پردازش شروع شود را زمان انتظار در صف می‌نامیم میانگین تمام درخواست‌ها برای تمام نودها به عنوان میانگین زمان صف (q) همه درخواست‌ها تعریف می‌شود. این پارامتر کیفیت سرویس‌دهی سیستم را نشان می‌دهد و بر حسب ثانیه اندازه‌گیری می‌شود.

۳-۲- زمان پردازش (CET_t)

بنابراین کل زمان تأخیر بصورت رابطه زیر به دست می‌آید:

(۱۴)

$$Delayt = q + CETt + TR + TS$$

زمان پردازش وظایف تخصیص داده شده به نود FN_j از طریق رابطه زیر به دست می‌آید:

UNCORRECTED PROOF

(۱۵)

$$CET_t = \sum_{k=1}^{r} (R_i Task_k^j.StartTime + R_i Task_k^j.ExeTime)$$

(۱۶)

$$i \in jobs\ of\ selected\ tasks$$

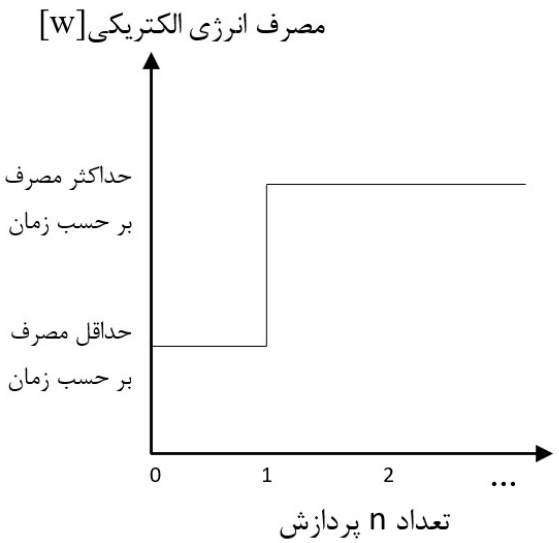
StartTime زمان شروع در پردازنده و ExeTime مدت زمانی را که طول می‌کشد وظیفه بر روی پردازنده اجرا شود، نشان می‌دهند این مقدار از تقسیم حجم بر روی قدرت پردازشی پردازنده به دست می‌آید.

زمان ارسال (T_S) و دریافت نتایج (T_R):

زمان ارسال (T_S) و دریافت نتایج (T_R) بر اساس تقسیم حجم وظیفه بر پهنای باند محاسبه می‌شود و در نهایت میانگین بین تمام مقادیر محاسبه می‌شود.

در هر فاگ بود نه تنها زمان اجرای درخواست مهم است بلکه انرژی مصرف شده نیز اهمیت دارد. از این‌رو مدل‌های مختلف مصرف انرژی تا کنون پیشنهاد شده است تا مقدار برق مصرفی برای انجام فرایندها را نشان دهد.

در این مقاله مدل ساده مصرف انرژی را در نظر می‌گیریم ۱۷ در اینجا، مصرف انرژی (PC [W]) از یک گره فاگ FN حداکثر XE_{FN} [W] است اگر حداقل یک فرآیند انجام شود، در غیر این صورت PC حداقل mE_{FN}[W] است که در شکل ۴ نشان داده شده است.



شکل ۴- **مدل ساده مصرف انرژی** [۱۷]

مصرف انرژی الکتریکی EC(x) [J] یک گره فاگ FN برای پردازش داده‌های ورودی با اندازه x و Admin برای داده Z به شرح زیر است:

(۱۷)

$$EC_{FN} = xE_{FN}[W].EC_{Admin}(x)$$

(۱۸)

$$E_{Admin} = xCET_{Admin}[W].CET_{Admin}(z)$$

z نشان دهنده اطلاعاتی است که به گره ادمین ارسال می‌شود. این اطلاعات شامل اطلاعات حجم داده (کارها) ID کاربر میزان مصرف انرژی توسط نودهای فاگ و سرعت پردازنده‌ها در فاگ تودها است. مصرف برق الکتریکی PI و

UNCORRECTED PROOF

PO از ماژول‌های ورودی ۱ و خروجی ۰ متناسب با نرخ دریافت و انتقال از کاربر u، گره فاگ FN و ادمین admin است.[۲۴] در اینجا چهار نوع انتقال داریم، انتقال از کاربر به نود فاگ $TI_{u\to FN_j}$ و بلعکس $TO_{u\to FN_j}$، انتقال از نود فاگ به ادمین $TI_{FN_j\to Admin}$و بلعکس $TO_{FN_j\to Admin}$ اگر x را داده و z را اطلاعات داده در نظر بگیریم بنابراین مصرف انرژی الکتریکی برای دریافت و ارسال داده x و z در هر سطح بصورت زیر محاسبه می شود.

(۱۹)

$$EI_{u\to FN_j} = PI_{u\to FN_j}.TI_{u\to FN_j}(x)$$

(۲۰)

$$EI_{FN_j\to \text{Admin}} = PI_{FN_j\to \text{Admin}}.TI_{FN_j\to \text{Admin}}(x+z)$$

(۲۱)

$$EO_{u\to FN_j} = PO_{u\to FN_j}.TO_{u\to FN_j}(x)$$

$$EO_{FN_j\to \text{Admin}} = PO_{FN_j\to \text{Admin}}.TO_{FN_j\to \text{Admin}}(x+z)$$

PS [W] نشان دهنده توان الکتریکی است که در نود فاگ با PS_{FN} و در ادمین با PS_{Admin} نشان داده می‌شود و میزان مصرف انرژی برای ذخیره‌سازی داده را نشان می‌دهد. این پارامتر به نرخ دسترسی یا a [bps] بستگی دارد. بنابراین مصرف انرژی الکتریکی ES[J] برای ذخیره‌سازی داده ورودی و داده خروجی در نود فاگ و ادمین برای داده با حجم X بصورت زیر محاسبه می‌شود.

(۲۲)

$$ES_{u\to FN}(x) = PS_{u\to FN}.TS_{u\to FN}(x)$$

(۲۳)

$$ES_{FN\to \text{Admin}}(x) = PS_{FN\to \text{Admin}}.TS_{FN\to \text{Admin}}(x)$$

(۲۴)

$$ES_{FN\to u}(x) = PS_{FN\to \text{u}}.TS_{FN\to \text{u}}(x)$$

(۲۵)

$$ES_{\text{Admin}\to FN}(x) = PS_{\text{Admin}\to FN}.TS_{\text{Admin}\to FN}(x)$$

فرض کنیم کاربر k، کار i را ارسال می‌کند. این کاربر به فاگ نود و دسترسی دارد. اگر سایر فاگ نودها را با FN نشان دهیم در این صورت مصرف کل انرژی در تمام سیستم از شروع درخواست تا پایان به صورت زیر محاسبه می‌شود.

(۲۶)

$$Total_{Consume_{Energy}} = (EI+EO+ES)_{u_i\to FN_j} + (EI+EO+ES)_{FN_j\to Admin} + EC_{Admin}$$

(۲۷)

$$+(EO+EI+ES)_{Admin\to GN} + EC_{FN} + (EI+EO+ES)_{FN\to Admin}$$

(۲۸)

$$+(EO+EI+ES)_{Admin\to FN_j} + (EI+EO+ES)_{FN_j\to u_i}$$

بنابراین در اینجا مسئله کمینه‌سازی زمان اجرای درخواست‌ها در محیط محاسباتی فاگ- ابر با جستجو در زیر مجموعه زیر تعریف می‌شود:

(۲۹)

$$FNTasks = \{FN_1Tasks, FN_2Tasks, \dots, FN_mTasks\}$$

در اینجا:

(۳۰)

$$FN_jTasks = \{J_aTasks^i_x, J_bTasks^i_y, \dots, J_iTasks^i_k, \dots, J_nTasks^i_r\}$$

تابع هزینه در اینجا یک تابع هزینه برای تعیین کیفیت راه حل مورد انتظار (FNTasks) تعریف شده است.

(۳۱)

$$Cos_function([FNTasks]) = \min\left[\sum_{j=1}^{m}\left(Cos_{function}\left(J_iTasks^i_k,[FN_j]\right)\right)\right]$$

این تابع هزینه یک تابع کمینه‌سازی برای اندازه‌گیری بهینه بودن دو هدف تأخیر و انرژی مصرف شده به شرح زیر است.

UNCORRECTED PROOF

(۳۲)

$$Cos_{function}([FNTasks]) = k_1.\text{Delay} + k_2\text{Total_Consume_Energy} + k_1 + k_2 = 1$$

در اینجا k_1 و k_2 فاکتورهای وزن هستند که برای تأکید بر اهمیت هر یک از دو هدف ارزیابی (یعنی زمان اجرای CPU و انرژی مصرف شده) ثابت در نظر گرفته شده‌اند. به عبارت دیگر انتخاب وزن در چنین رویکردهای بهینه‌سازی چند هدف منعکس کننده ترجیح تصمیم گیرنده است.

۳-۳- حل مسئله

جدول ۳ مقایسه عملکرد افراد گروه G1 در انجام وظایف مختلف طی دو تکرار (R1، R2)

	۴	۲	۳	۱	۴	۲
G1	Task1	Task2	Task3	Task1	Task2	Task3
	R1			R2		

تعداد ژنوم‌های موجود در گیاه در اینجا برابر N است با توجه به هورمون‌های SA، JA و ET کل جمعیت‌ها را به سه دسته تقسیم‌بندی می‌کنیم فرآیند الگوریتم دفاع گیاهان بصورت زیر است:

۱-برای تقسیم‌بندی جمعیت از الگوریتم K میانگین استفاده می‌شود.

۲-بعد از خوشه‌بندی خوشه مربوط به هورمون SA با استفاده از رابطه ۱ بروز رسانی می‌شود.

۳-سپس بر اساس رابطه ۲ بخشی از جمعیت خوشه JA با خوشه SA ادغام می‌شود.

۴-سپس بخشی از خوشه JA با ET بر اساس رابطه (۳) ادغام می‌شود و برعکس بر اساس رابطه (۴)

۵-در ادامه δ درصد از اعضای جامعه بر اساس رابطه ۵ تخریب می‌شوند.

۶- در پایان اعضا خوشه ET با استفاده از رابطه ۶ افزایش می‌یابد.

۷-برازندگی هر هورمون یا جواب با استفاده از رابطه ۷ محاسبه می‌شود.

۸- نرخ رشد μ در جمعیت اعمال می‌شود و مازاد بر اساس کمترین برازندگی حذف می‌شوند.

۹-این فرآیند تا رسیدن به جواب ادامه پیدا می‌کند.

الگوریتم بهینه‌سازی دفاع گیاهان برای مدل پیشنهادی ارائه شده است تا تأخیر را به حداقل برساند علاوه بر این الگوریتم بهینه‌سازی دفاع گیاهان شامل روشی برای جریمه کردن راه‌حل‌های احتمالی است که محدودیت‌های مسئله را برآورده نمی‌کند. این امر باعث می‌شود که برای تولید جواب‌های جدید احتمال انتخاب جواب‌های غیر ممکن کمتر باشد.

۴- ارزیابی

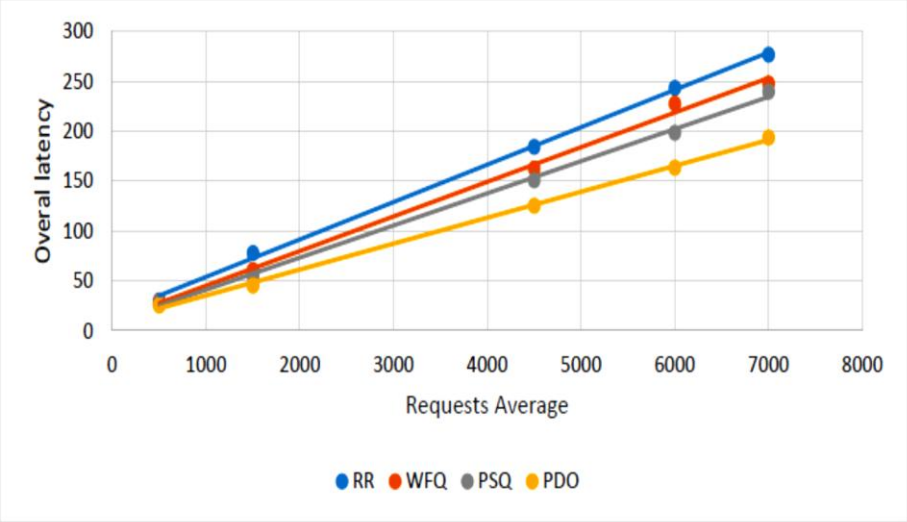
تمام آزمایشات روی یک رایانه رومیزی با یک پردازنده چهار هسته‌ای سرعت ۲۶ گیگاهرتز و ۸ گیگابایت حافظه رم انجام شد.

این آزمایش شامل مجموعه‌ای از ۱۶ سرور با سرعت پردازش متوسط ۵۰۰ بسته در ثانیه و به طور گسترده از ۵۰ تا ۱۰۰۰ بسته در ثانیه توزیع می‌شود. میانگین تأخیر سرورها به طور متوسط ۵ میلی ثانیه در هر بسته تنظیم شده است. همچنین از ۱ میلی ثانیه تا ۹.۷ میلی ثانیه توزیع می‌شود در مجموع از

از آنجا که کاهش تأخیر خدمات در یک محیط یک مسئله بهینه‌سازی است. می‌توان آن را با استفاده از الگوریتم بهینه‌سازی مانند PMO حل کرد ما فرض می‌کنیم که یک درخواست شامل زیر مجموعه‌ای از وظایف است. به عنوان مثال در اینجا ما دو درخواست داریم که هر کدام از آن‌ها از سه وظیفه تشکیل شده است. از طرف دیگر فرض کنید که چهار گره مه در سیستم وجود دارد. بنابراین هر هورمون در گیاه در اینجا یک آرایه ۶ بعدی است هر آرایه یک وظیفه را نشان می‌دهد شماره اختصاص داده شده به هر آرایه نشان دهنده گره مه شاخص است. هورمون گیاه با پاسخ مسئله در زیر نشان داده شده است:

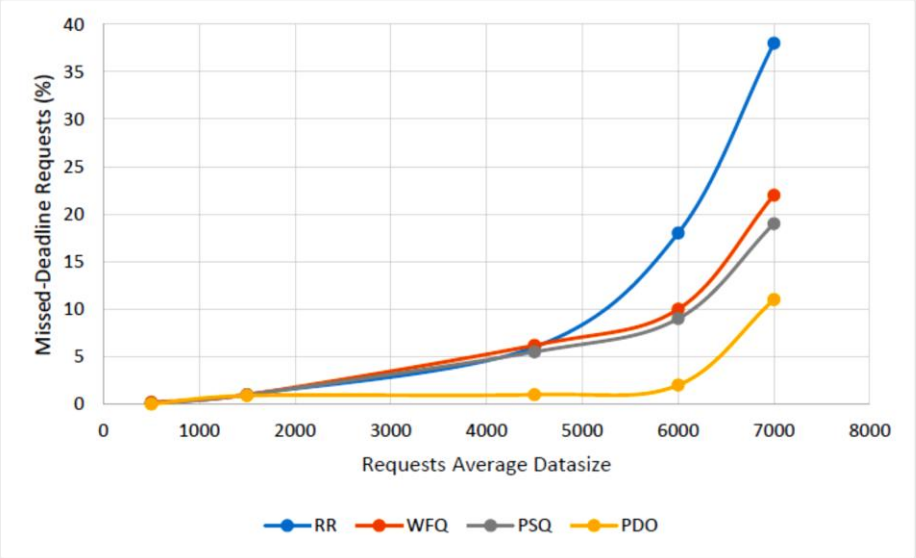
UNCORRECTED PROOF

۱۰۰ درخواست در آزمایش استفاده شده است. اولویت‌ها به طور یکنواخت از ۱ تا ۱۶ توزیع می‌شوند. الزامات مهلت اجرا به طور متوسط ۴۰۰ ثانیه با واریانس ۵۰ تنظیم شده است. در ابتدا میانگین اندازه داده‌ها از یک مقدار کوچک شروع می‌شود بطوریکه مهلت درخواست‌ها را کاهش می‌دهد. هیچ درخواستی مهلت اجرا خود را از دست نمی‌دهد. سپس میانگین اندازه داده افزایش می‌یابد تا تأثیر آن بر تأخیر کلی و تعداد درخواست‌های از دست رفته مشاهده شود. شکل‌های ۵ و ۶ میانگین تأخیر کلی و تعداد درخواست‌های از دست رفته در مقابل اندازه متوسط درخواست‌ها را نشان می‌دهد. الگوریتم بهینه‌سازی دفاع گیاهان نسبت به بقیه الگوریتم‌ها تأخیر کلی کمتری را نشان می‌دهد. نتایج WFQ و PSQ بسیار نزدیک به یکدیگر هستند زیرا تخصیص درخواست‌ها در منابع در هر دوی این الگوریتم‌ها بر اساس اولویت به دست می‌آید با این حال ارسال متفاوت است. RR بالاترین زمان تأخیر را نشان می‌دهد و دلیل آن نحوه تخصیص درخواست‌ها به روش RR بدون در نظر گرفتن اولویت‌های درخواست است. همچنین می‌توان مشاهده کرد که الگوریتم بهینه‌سازی دفاع گیاهان رکورد بدون درخواست‌های از دست رفته را برای مدت طولانی‌تری نسبت به سایر الگوریتم‌ها نگه می‌دارد. با این حال در اندازه متوسط ۶۵۰۰ بسته الگوریتم بهینه‌سازی دفاع گیاهان نمی‌تواند تضمین کند. که تمام مهلت‌های درخواست‌ها برآورده می‌شود زیرا تأخیر سرویس آن‌ها افزایش می‌یابد و مهلت آن‌ها بحرانی می‌شود. ذکر این نکته حائز اهمیت است که در تمام داده‌ها با استفاده از الگوریتم بهینه‌سازی دفاع گیاهان قادر به ارائه یک‌راه حل زمان‌بندی عملی است که در آن تمام مهلت‌های درخواست‌ها باید رعایت شود. با این حال نتایج شبیه‌سازی نشان می‌دهد امکان برآورده کردن همه درخواست‌ها وجود ندارد این موضوع در شکل ۶ بین ۶۰۰۰ و ۸۰۰۰ در اندازه متوسط داده نشان داده شده است. پس از ۸۰۰۰ بسته یافتن جواب غیر ممکن می‌شود و از این‌رو نمی‌توان یک برنامه زمان‌بندی عملی پیدا کرد. این موضوع در شبیه‌سازی نشان داده شده است زیرا تأخیر واقعی هر بسته می‌تواند با تأخیر متوسط متفاوت باشد.



شکل ۵ - **میانگین تأخیر کلی در برابر حجم داده**

UNCORRECTED PROOF



شکل ۶- **تعداد درخواست‌های از دست رفته در مقابل اندازه متوسط درخواست‌ها**

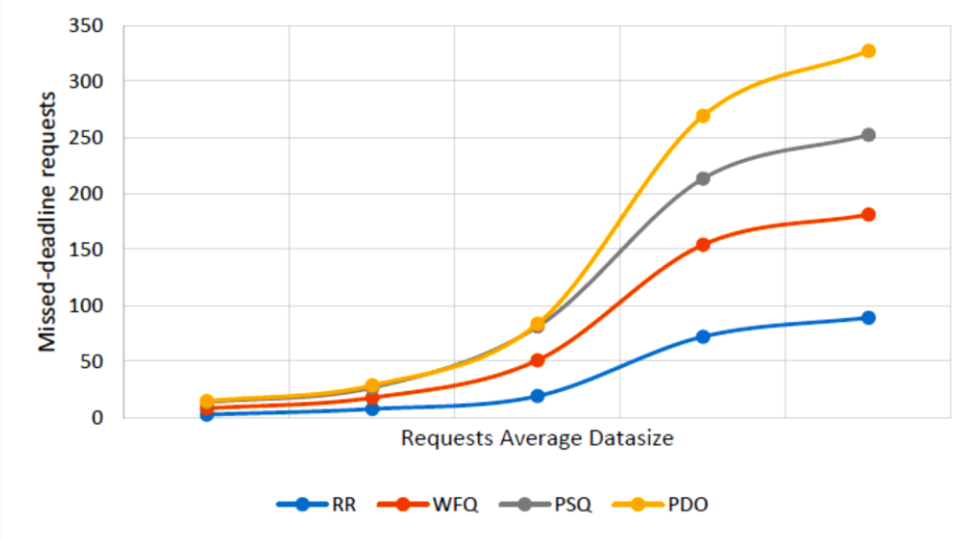
۴-۱- برنامه‌ریزی پویا

در آزمایش دیگر ۱۶ سرور و ۵۰۰ درخواست با هدف ارزیابی تأخیر دررسیدن درخواست‌ها طراحی شد. درخواست‌ها بصورت تصادفی با توزیع پواسون با میانگین ورود ۱ ثانیه ایجاد می‌شوند. اولویت‌ها بین ۱ تا ۱۶ با توزیع یکنواخت تعیین شده است الزامات ضرب‌الاجل به طور متوسط روی ۲۰۰ ثانیه با واریانس ۵۰ تنظیم شده است. بازه زمانی که در آن منابع برنامه‌ریزی می‌شوند ۱۰ ثانیه است. متوسط اندازه درخواست معمولاً از ۱۰۰۰ تا ۱۰۰۰۰ بسته است. شکل‌های ۷ و ۸ میانگین کلی تأخیر و تعداد درخواست‌های از دست رفته را در مقابل میانگین اندازه داده‌ها نشان می‌دهد. الگوریتم بهینه‌سازی دفاع گیاهان بهترین تأخیر کلی را در مقایسه با سایر الگوریتم‌ها نشان می‌دهد. نتایج WFQ و PSQ بسیار نزدیک به یکدیگر هستند با این حال این تفاوت با افزایش میانگین اندازه داده‌ها افزایش می‌یابد. از سوی دیگر شکل ۸ نشان می‌دهد که اگر اندازه داده به طور متوسط کمتر از ۵۰۰۰ بسته باشد. الگوریتم بهینه‌سازی دفاع گیاهان تقریباً تمام درخواست‌ها را می‌تواند پردازش کند با اندازه متوسط داده ۶۰۰۰ بسته الگوریتم بهینه‌سازی دفاع گیاهان درخواست از دست رفته‌ای ندارد WFQ و PSQ تقریباً ۱۰ از درخواست‌ها و RR از درخواست‌ها را از دست داده‌اند. پس از این مرحله اکثر درخواست‌ها بسیار بحرانی و برخی از آن‌ها حتی غیر قابل اجرا می‌شوند. در الگوریتم بهینه‌سازی دفاع گیاهان پس از افزایش اندازه داده به بالای ۶۰۰۰ بسته درخواست‌های از دست رفته شروع می‌شود. با این حال هنوز نتایج بهتری نسبت به ۳ الگوریتم دیگر نشان می‌دهد.

UNCORRECTED PROOF



شکل ۷ - **میانگین کلی تأخیر در مقابل حجم داده**



شکل ۸ - **تعداد درخواست‌های از دست رفته را در مقابل میانگین حجم داده‌ها**

۴-۲- مقایسه معماری‌های ابری و ترکیبی ابر مه

هدف این بخش ارزیابی تأخیر سرویس ارائه شده توسط منابعی است که دارای ویژگی‌های محاسبات ابری و مه هستند. به طور کلی منابع ابری به عنوان منابعی قدرتمند با قابلیت پردازش بالا طبقه‌بندی می‌شوند، اما در عین حال میانگین انتقال و تأخیر شبکه‌ای زیادی دارند ،برعکس منابع مه قدرت پردازش محدودی دارند. اما از آنجایی که نزدیک به لبه هستند میانگین تأخیر کمتری را ارائه می‌کنند در اینجا آزمایشی طراحی شده است تا مشخص شود که بهتر است منابع بسیار قدرتمند در ابر مورد استفاده قرار بگیرند یا منابع با قدرت کمتر در مه تعبیه شوند. برای اینکه بتوان به مجموعه‌ای از درخواست‌ها را با کمترین تأخیر پاسخ داد. در مدل فرموله شده سه پارامتر به شرح زیر در نظر گرفته شده است.

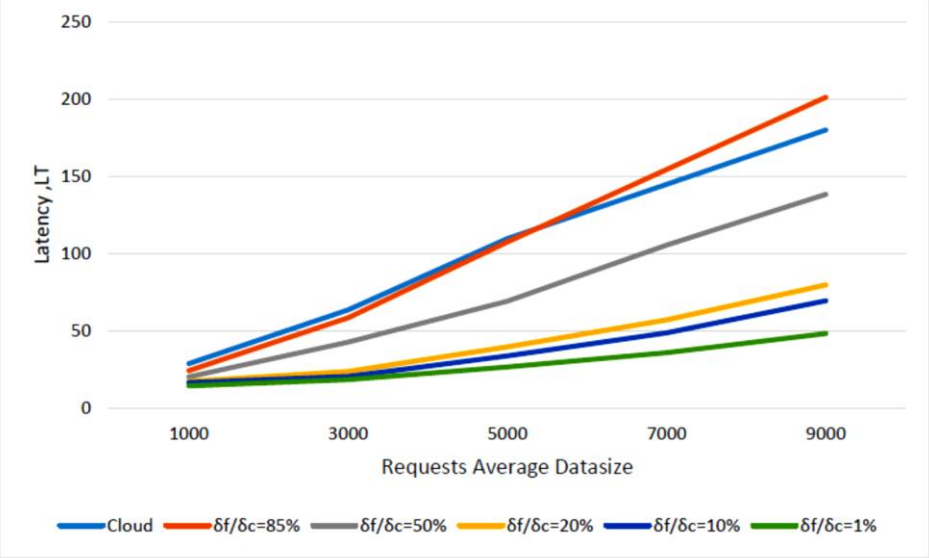
- نسبت میانگین تأخیر $\overline{\delta f}/\overline{\delta c}$
- نسبت سرعت پردازش Pf/Pc
- نسبت تعداد منابع Nf/Nc

تأثیر هر پارامتر بر تأخیر سرویس به طور مستقل با ثابت کردن دو مورد از آن‌ها و تغییر تنها یکی مورد مطالعه قرار می‌گیرد. تأخیری که با تغییر این پارامترها به دست می‌آید در برابر سیستمی که دارای ویژگی ابری است ارزیابی می‌شود. این

سیستم ابری دارای مجموعه ای از ۴ سرور ابری با قابلیت پردازش بسیار بالا ۵۰۰۰ بسته در ثانیه است با این حال میانگین تأخیر برای این سرورها تا ۱۰ میلی ثانیه در هر بسته تعیین شده است. برای ارزیابی تأخیر در مجموع از ۵۰۰ درخواست در این مجموعه آزمایش‌ها استفاده می‌شود. ورود آن‌ها از توزیع پواسون با میانگین زمان ورود ۱ ثانیه پیروی می‌کند تأخیر در مقایسه با اندازه متوسط داده مورد مطالعه قرار می‌گیرد که از ۱۰۰۰ بسته به ۱۰۰۰۰ بسته تغییر می‌کند.

۴-۳- تأثیر نسبت میانگین تأخیر

در این آزمایش تعداد سرورهای مه چهار برابر تعداد سرورهای ابری Nf/Nc=4 است. قدرت پردازش آن‌ها تنها ۱۰٪ ابر Pf/Pc=10% است. میانگین نسبت تأخیر $\delta f/\delta c$ ، به ۱٪، ۱۰٪، ۲۰٪، ۵۰٪ و ۸۵% تغییر می‌کند شکل ۹ نتایج تأخیر را نشان می‌دهد و همان‌طور که مشاهده می‌شود افزایش میانگین نسبت تأخیر از ۱٪ به ۸۵٪ تأخیر را افزایش می‌دهد تا زمانی که به نقطه‌ای برسد که از تأخیر ابری عبور کند. این موضوع نشان می‌دهد که سرورهای مه از دستگاه‌های لیبه دور هستند و به سرورهای ابری نزدیک‌تر هستند.

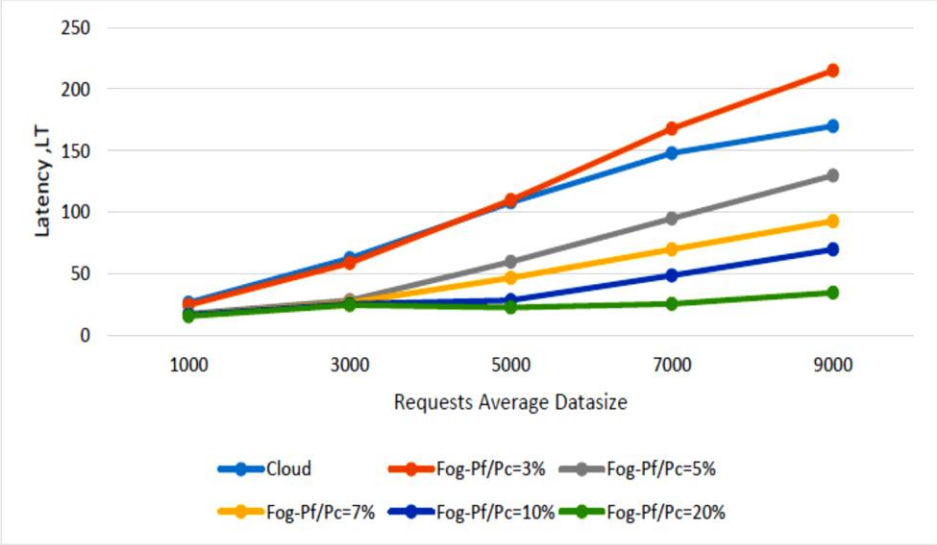


شکل ۹- **تأخیر بود فاگ نسبت به ابر**

۴-۴- تأثیر نسبت سرعت پردازش

در این آزمایش تعداد سرورهای مه چهار برابر تعداد سرورهای ابری، Nf/Nc تنظیم شده است. میانگین تأخیر آن‌ها $10\ \delta f/\delta c$ درصد است. قدرت پردازش Pf/Pc، به ۳، ۵، ۱۰ و ۲۰ درصد تغییر می‌کند. شکل ۱۰ نتایج تأخیر را نشان می‌دهد. همان‌طور که نتایج ارائه شده در شکل ۱۰ نشان می‌دهد. کاهش نسبت سرعت پردازش از ۲۰٪ به ۳٪ تأخیر مه را افزایش می‌دهد تا زمانی که به نقطه‌ای برسد که از تأخیر ابری عبور کند. این موضوع به‌واسطه وجود سرورهای مه با قابلیت پردازشی کم ایجاد می‌شود. در این حالت حتی اگر این منابع نسبت به منابع ابری به لبه نزدیک‌تر باشند مه به دلیل قابلیت‌های پردازش کم تأخیر بالایی ارائه می‌کند.

UNCORRECTED PROOF

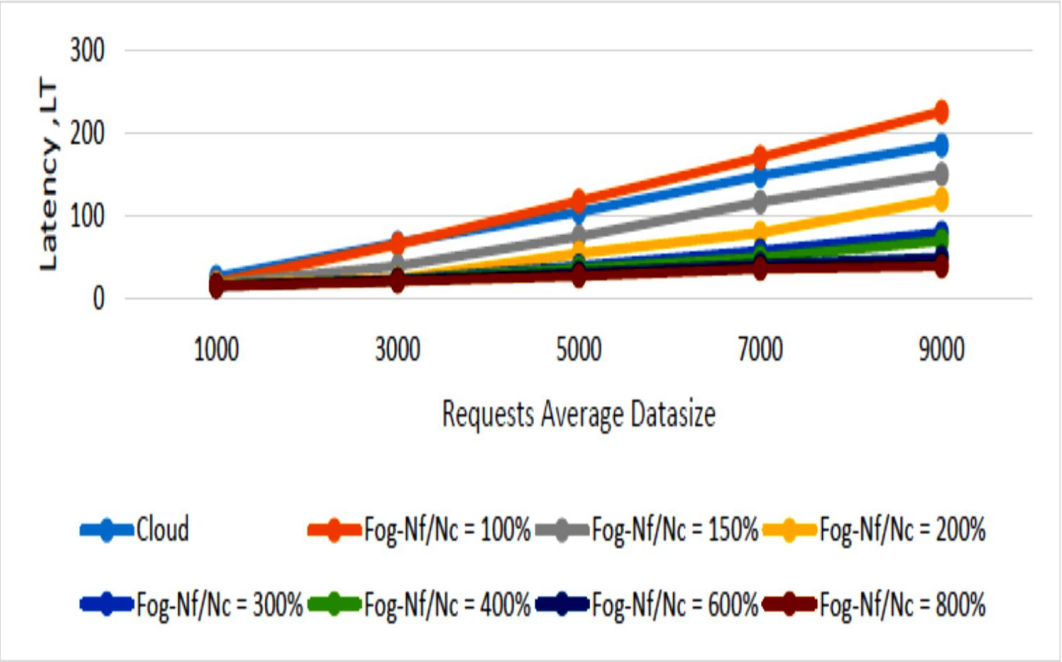


شکل ۱۰- **تأخیر فاگ نسبت به ابر با تغییر سرعت پردازش در لایه فاگ**

۴-۵- تأثیر نسبت تعداد منابع

در آزمایش ارائه شده در شکل ۱۱ میانگین تأخیر سرورهای مه بر روی ۱۰ درصد تأخیر میانگین ابر تنظیم شده است. همچنین توان پردازشی سرورهای مه روی ۱۰ درصد از قابلیت‌های سرورهای ابری تنظیم شده است. تعداد منابع مه نسبت به ابری به ۱۰۰٪، ۱۵۰٪، ۲۰۰٪، ۳۰۰٪، ۴۰۰٪، ۶۰۰٪، و ۸۰۰٪، تغییر می‌کند. شکل ۱۱ نتایج تأخیر را برای این آزمایش نشان می‌دهد. همان‌طور که از آزمایش ارائه شده در شکل ۱۱ مشخص است کاهش نسبت تعداد منابع از ۸۰۰ درصد به ۱۰۰ درصد تأخیر مه را افزایش می‌دهد تا زمانی که به نقطه‌ای برسد که از تأخیر ابری عبور کند. این موضوع به خاطر داشتن سرورهای مه با منابع بسیار کم است. حتی اگر این منابع به لبه نزدیک‌تر باشند و قابلیت پردازش خوبی داشته باشند، داشتن منابع کمتر بر تأخیر تأثیر منفی می‌گذارد.

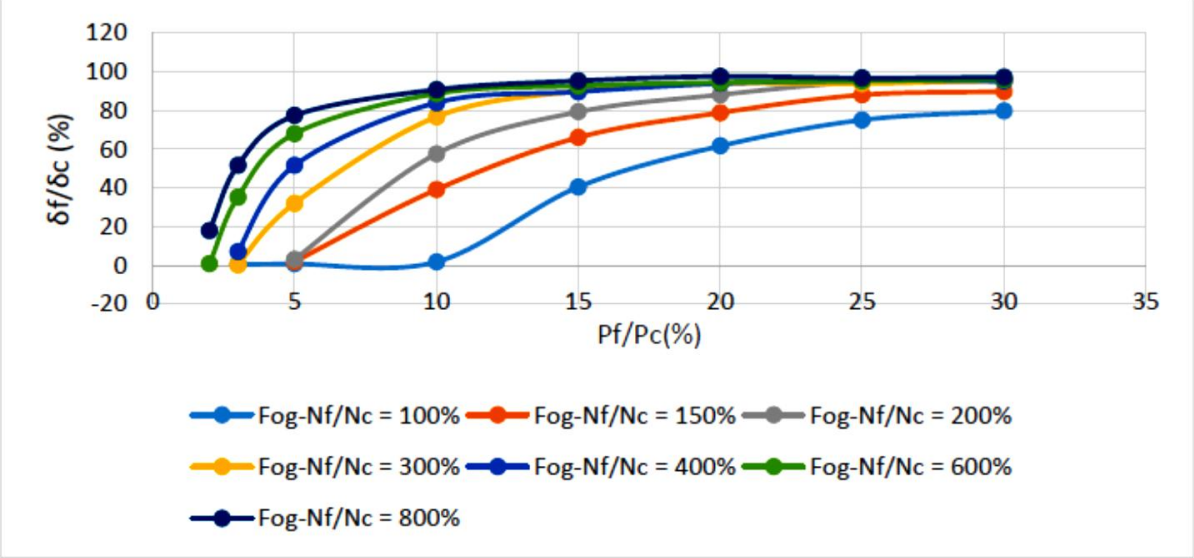
UNCORRECTED PROOF



شکل ۱۱ - **تأخیر نود فاگ در مقایسه با کلود با تغییر تعداد سرور در لایه فاگ**

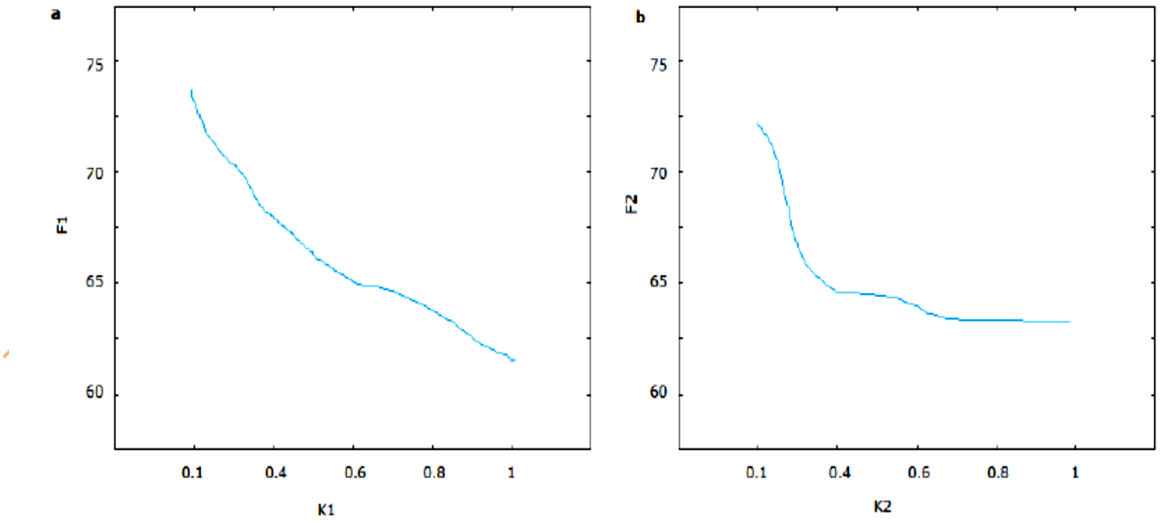
آزمایش دیگری با هدف یافتن نقاط شکست از نظر سه پارامتر میانگین تأخیر قدرت پردازش و تعداد منابع انجام شده است. اندازه متوسط داده‌ها روی ۵۰۰۰ بسته تنظیم شده است. شکل ۱۲ نتایج آزمایش را به عنوان نقاط تلاقی بین تأخیر مه و تأخیر ابر نشان می‌دهد. آزمایش نشان می‌دهد که در آن مجموعه ای از سرورهای مه نسبت به مجموعه ای از سرورهای ابری تأخیر بهتری را ارائه می‌دهند به عنوان مثال برای یک نسبت تأخیر متوسط خاص و نسبت قدرت پردازش بین مه و ابری نمودار تعداد نسبت سرورهایی را نشان می‌دهد که در آن مه بهتر از ابری عمل می‌کند یا بلعکس در سه آزمایش اخیر مشاهده شد. که تأخیر محیط مه در نقطه‌ای به دلیل تأخیر بالا (شکل (۹) قدرت پردازشی کم (شکل (۱۰) و تعداد کم سرورهای مه شکل (۱۱) از تأخیر محاسبات ابری عبور می‌کند.

UNCORRECTED



شکل ۱۲- **نقاط شکست فاگ و محاسبات ابری**

شکل ۱۳ تغییر مقادیر K وزن هر بخش از تابع هدف را نشان می‌دهد شکل ۱۳- تغییرات میزان تأخیر را با تغییر وزن آن در تابع هدف نشان می‌دهد همان‌طور که نشان داده شده است با افزایش میزان وزن آن مقدار تابع F^1 (میزان تأخیر) نیز کاهش می‌یابد. این موضوع کاملاً مشخص است زیرا هر چه وزن افزایش پیدا می‌کند میزان جریمه نیز افزایش پیدا کرده الگوریتم به سمت تک بعدی شدن حرکت می‌کند بنابراین انتظار می‌رود به ازای یک بخش از تابع عملکرد بهتری داشته باشد. شکل ۱۳-b رفتار مشابهی را در ازای بخش دوم تابع یعنی مصرف انرژی نشان می‌دهد. در اینجا با افزایش وزن k^2 مقدار بخش دوم تابع هدف کاهش پیدا کرده است و سپس ثابت شده است.



شکل ۱۳- **تغییر مقادیر K بر روی هر بخش از تابع هدف**

۴-۶- نمونه مثال عددی

در ابتدا فرض می‌شود که ۵ نود فاگ برای تخصیص ۱۰ درخواست جهت پردازش در اختیار داریم جدول زیر ده ژن را نشان می‌دهد که بصورت تصادفی و با توزیع نرمال ایجاد شده‌اند. هر زن ۱۰ بعد دارد که نشان دهنده ده درخواست است. از آنجایی که الگوریتم بهینه‌سازی دفاع گیاهان پیوسته است و مسئله تخصیص درخواست گشته است در اینجا از تابع جزء صحیح استفاده می‌شود.

جدول ۴ نتایج اندازه‌گیری ویژگی مورد نظر در ده گروه گیاهی تحت ده تیمار مختلف (T1 تا T10)

Plant	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
G1	۴.۲۴	۲.۴۳	۱.۲۹	۳.۳۶	۴.۶۴	۱.۷۸	۲.۷۳	۴.۰۰	۱.۱۶	۴.۷۹
G2	۴.۰۵	۳.۲۴	۱.۷۴	۲.۹۹	۳.۰۷	۴.۹۸	۴.۴۲	۴.۸۵	۳.۷۲	۲.۶۱
G3	۴.۴۷	۲.۹۲	۱.۹۳	۲.۵۹	۳.۸۲	۳.۲۳	۴.۰۳	۴.۹۸	۴.۸۵	۳.۱۴
G4	۴.۸۹	۱.۴۶	۱.۲۱	۲.۲۲	۳.۳۲	۳.۱۲	۴.۶۰	۳.۱۶	۲.۷۳	۳.۱۷
G5	۳.۸۵	۱.۰۷	۴.۲۰	۱.۵۷	۲.۹۱	۲.۰۳	۲.۴۸	۳.۶۵	۱.۶۸	۲.۱۲
G6	۱.۷۹	۱.۷۸	۲.۳۱	۴.۵۲	۲.۸۸	۲.۶۲	۱.۷۲	۴.۸۸	۲.۶۳	۴.۳۸
G7	۳.۴۶	۲.۵۱	۴.۵۱	۴.۱۴	۲.۸۶	۴.۲۶	۴.۵۹	۲.۷۲	۲.۳۴	۳.۳۹
G8	۴.۶۱	۳.۸۱	۲.۵۱	۳.۹۴	۴.۸۲	۳.۱۷	۳.۱۶	۲.۲۴	۱.۲۸	۱.۷۳
G9	۱.۳۷	۲.۸۵	۱.۰۴	۴.۶۶	۳.۵۷	۱.۰۱	۱.۱۲	۱.۸۳	۲.۸۲	۱.۵۱
G10	۱.۰۳	۳.۹۱	۲.۴۲	۴.۱۲	۲.۷۵	۲.۷۵	۱.۲۰	۱.۲۰	۱.۳۶	۳.۳۸

همان‌طور که گفته شد برای تخصیص هر ژن به هر گروه از خوشه‌بندی استفاده می‌شود. فرض کنیم نتایج خوشه‌بندی بصورت زیر می‌باشد.

جدول ۵ گروه‌بندی ده رقم گیاهی (G1 تا G10) بر اساس عوامل ET ، JA و SA

SA				JA			ET		
G1	G2	G3	G4	G5	G6	G7	G8	G9	G10

بر اساس میزان پارامترهای α و β و اعداد تصادفی ایجاد شده ژنها مربوط به گروه SA تغییر پیدا می کنند به عنوان مثال معادله ۱ بر روی G^1 اعمال می شود :

$$G = \begin{bmatrix} 1.37 \\ 2.85 \\ \dots \\ 2.82 \\ 1.51 \end{bmatrix} + r \times w \times \left(\begin{bmatrix} 1.37 \\ 2.85 \\ \dots \\ 2.82 \\ 1.51 \end{bmatrix} - \begin{bmatrix} 4.24 \\ 2.43 \\ \dots \\ 1.16 \\ 4.79 \end{bmatrix} \right) = \begin{bmatrix} 1 \\ 2.36 \\ \dots \\ 2.18 \\ 1.45 \end{bmatrix} \qquad (۳۳)$$

در ادامه ژن های مربوط به خانواده JA با خانواده SA ترکیب میشوند در اینجا فرض می شود G^5 با G^1 انتخاب شده‌اند و بصورت زیر با هم ادغام می‌شوند.

UNCORRECTED

$$G5_{JA} = \begin{bmatrix} 3.85 \\ 1.07 \\ \dots \\ 1.68 \\ 2.12 \end{bmatrix} + \vec{r} \times w_4 \times \left(\begin{bmatrix} 1 \\ 2.36 \\ \dots \\ 2.18 \\ 1.45 \end{bmatrix} - \begin{bmatrix} 3.85 \\ 1.07 \\ \dots \\ 1.68 \\ 2.12 \end{bmatrix} \right) = \begin{bmatrix} 2.18 \\ 3.28 \\ \dots \\ 2.84 \\ 0.97 \end{bmatrix} \qquad (۳۴)$$

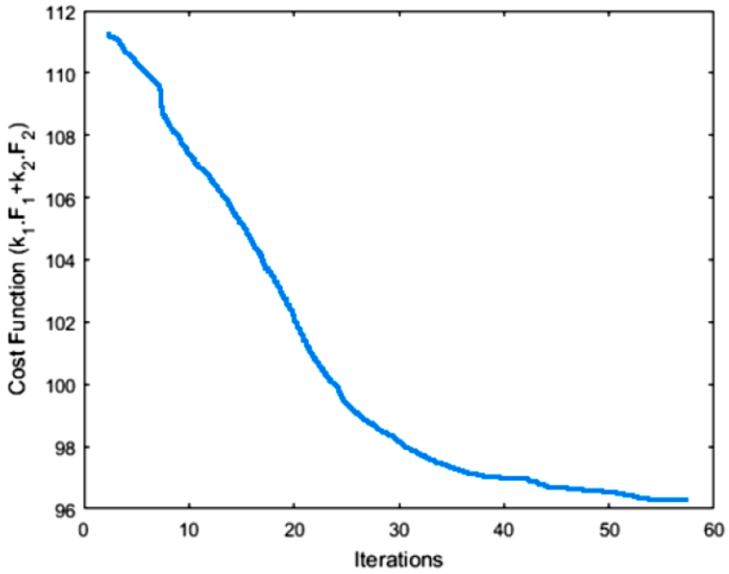
به همین ترتیب فرایندهای ادغام را بر روی سایر زن‌ها اعمال می شود. در نهایت برازندگی تمام زن ها محاسبه می شود برای محاسبه برآزندگی در ابتدا باید قدر مطلق ارائه‌های هر

۲ ۳ ۱ ۳ ۵ ۲ ۳ ۴ ۳ ۱ (۳۵)

ماتریس ۳۵ نشان می‌دهد که وظیفه اول به فاگ نود۲، وظیفه دوم به فاگ نود ۳ و .... تخصیص داده می شود. نتایج تغییرات

ژن محاسبه شود به عنوان مثال برای ژن G_{5Ja} به دست می‌آید:

برازندگی در تعداد نسل‌های مختلف در شکل ۱۴ نشان داده شده است.



شکل ۱۴- **تغییرات برازندگی با نسل**

۴-۷- تحمل پذیری خطا و تاثیر آن بر تابع هزینه

تحمل خطا فرآیندی است که یک سیستم عامل را قادر می‌سازد تا به یک خطا در سخت‌افزار یا نرم‌افزار پاسخ دهد. این تعریف تحمل خطا به توانایی سیستم برای ادامه کار با وجود خطاها یا نقص‌ها اشاره دارد. سیستم عاملی که تعریف دقیقی از خطاها ارائه می‌دهد، نمی‌تواند توسط یک نقطه خطا مختل شود. این امر تداوم کسب‌وکار و در دسترس بودن بالای برنامه‌ها و سیستم‌های حیاتی را صرف نظر از هرگونه خطا تضمین می‌کند. تحمل خطا به جنبه‌هایی مانند متعادل‌سازی بار و بازیابی خطا وابسته است که خطر یک نقطه خطا را از بین می‌برد. این معمولاً بخشی از رابط سیستم عامل خواهد بود که به برنامه‌نویسان امکان می‌دهد عملکرد داده‌ها را در طول یک تراکنش بررسی کنند. انواع دیگر سیستم‌های تحمل خطا در صورت بروز برخی خطاها، افت عملکرد تدریجی را

UNCORRECTED PROOF

تجربه می‌کنند. این بدان معناست که تأثیر خطا بر عملکرد سیستم متناسب با شدت خطا است. به عبارت دیگر، یک خطای کوچک فقط تأثیر کمی بر عملکرد سیستم خواهد داشت تا اینکه باعث خطا کل سیستم یا مشکلات عملکردی عمده شود. بزرگترین عیب اتخاذ رویکرد تحمل خطا، هزینه انجام این کار است. سازمان‌ها باید با دقت در مورد عناصر هزینه یک سیستم مقاوم در برابر خطا یا با قابلیت دسترسی بالا فکر کنند. سیستم‌های مقاوم در برابر خطا، سازمان‌ها را ملزم به داشتن نسخه‌های متعددی از اجزای سیستم برای تضمین افزونگی، تجهیزات اضافی مانند ژنراتورهای پشتیبان و سخت‌افزار اضافی می‌کنند. این اجزا نیاز به نگهداری و آزمایش منظم دارند. همچنین فضای ارزشمندی را در مراکز داده اشغال می‌کنند. ما زمان‌های رسیدن نمایی بین خطاها را فرض می‌کنیم و اینکه خطاها مستقل هستند. علاوه بر این، طرح مبتنی بر هزینه ما برای مدیریت خطاهای فرآیند و گره در خوشه‌ها طراحی شده است. به منظور ارائه تخمین‌های دقیق برای کل زمان اجرای مورد انتظار در مقابل خطاها، مدل هزینه ما به این فرض بستگی دارد که اگر یک خطا رخ دهد، واسطه‌ها از بین نمی‌روند. یعنی، طرح مبتنی بر هزینه ما فرض می‌کند که می‌توانیم یک تخصیص را همیشه از آخرین نتیجه میانی با موفقیت تحقق یافته، مجدداً راه‌اندازی کنیم. در نتیجه، اگر همه واسطه‌ها در یک رسانه ذخیره‌سازی تحمل‌پذیر خطا جداگانه تحقق یابند، مدل هزینه ما دقیق خواهد بود زیرا واسطه‌ها در صورت خطا میان تخصیص از بین نمی‌روند. در صورتی که نتایج میانی به دلیل خطا میان تخصیص از بین بروند، مدل هزینه ما که زمان اجرای یک تخصیص را در حالت خطا تخمین می‌زند، بسیار خوش‌بینانه خواهد بود. بر پایه هدف مقاله پایه یعنی زمان‌بندی درخواست‌ها برای کمینه کردن تأخیر در محاسبات مه-ابری و همچنین تابع هزینه، پیشنهاد می شود که با آنالیز عملکرد تحمل پذیری خطاهای احتمالی، هزینه و تاثیر آن بر تأخیر برای پردازش یک درخواست در فرایند کامل یک تراکنش در نظر گرفته می شود. علاوه بر تأخیر، تحمل‌پذیری خطا برای بهینه‌سازی یک الگوریتم جدید مبتنی بر دفاع گیاهان با نام الگوریتم PDO در نظر گرفته می شود. بر اساس ساختار این الگوریتم فضای نمونه همان جامعه تصادفی هدف است و هر عضو آن یک تراکنش تخصیص درخواست به منبع است که برای تمام اعضا مجموعه یادگیری شده‌اند. تابع هزینه برای نرمال سازی تحمل‌پذیری خطا در برابر میزان تأخیر سرویس به دست آمده از الگوریتم PDO با سایر تکنیک‌های زمان‌بندی مقایسه شده است. نتایج رویکرد پیشنهادی بهبود قابل توجهی را در نرمال سازی تحمل پذیری خطای بهتری نسبت به سایر الگوریتم‌ها نشان داد. نتایج نشان می‌دهد در محاسبات مه به طور کلی نودهای مه تأخیر ارتباطی کمی دارند. این قابلیت استفاده از منابع محاسباتی مه کم مصرف با اندازه کوچک را فراهم می‌کند و تأخیر سرویس بهتری را نسبت به استفاده از محاسبات ابری ارائه می‌دهد.

در این پژوهش، عملکرد روش پیشنهادی عمدتاً با الگوریتم‌های کلاسیک زمان‌بندی مانند WFQ، PSQ و RR سنجیده شده است. از آنجا که یکی از چالش‌های کلیدی در محاسبات ترکیبی مه-ابر، وقوع خطا در نودهای لبه و سرورهای مه است، مدل پایه با فرض در دسترس بودن همیشگی منابع عمل می‌کند و استراتژی اختصاص وظایف را بر همین اساس نمایش می‌دهد.

۴-۸- پارامترهای تحمل خط

از آنجایی که تحمل خطا در این تحقیق از نوع پیشگیرانه است پارامتر های محاسبه عبارتند از: میانگین زمان پاسخ، مقیاس‌پذیری، توان عملیاتی و قابلیت اطمینان. میانگین زمان بین خرابی‌ها یا (MTBF)، میانگین زمان بین خرابی‌های سیستم را نشان می‌دهد. همچنین MTBF بالاتر نشان‌دهنده تحمل خطای بهتر است زیرا سیستم می‌تواند مدت زمان بیشتری بدون تجربه خرابی کار کند.

در عین حال، مراحل شناسایی رویکرد تحمل‌پذیری خطا نیز عبارتند از:

- تشخیص خطا
- ارزیابی آسیب
- بازیابی خطا
- رفع خطا و ادامه‌ی سرویس‌دهی سیستم

✓تشخیص خطا

برای تحمل یک خطا، ابتدا باید آن را تشخیص داد. از آنجایی که حالت‌های داخلی اجزا معمولاً قابل دسترسی نیستند،

UNCORRECTED PROOF

نمی‌توان یک خطا را مستقیماً تشخیص داد و از این رو، باید تظاهرات آن، که باعث می‌شوند سیستم به حالت خطا برود، تشخیص داده شود. بنابراین، نقطه شروع معمول برای تکنیک‌های تحمل خطا، تشخیص خطاها است.

✓ارزیابی آسیب

قبل از هرگونه تلاشی برای مقابله با خطای شناسایی شده، معمولاً لازم است میزان آسیب یا خراب شدن حالت سیستم ارزیابی شود. اگر تأخیر، که به عنوان فاصله تأخیر آن خطا شناخته می‌شود، بین بروز یک خطا و تشخیص پیامدهای نادرست آن زیاد باشد، احتمالاً آسیب به حالت سیستم گسترده‌تر از زمانی خواهد بود که فاصله تأخیر کوتاه‌تر باشد.

✓بازیابی خطا

پس از تشخیص خطا و ارزیابی خسارت، باید از تکنیک‌های بازیابی خطا برای دستیابی به حالت عادی سیستم بدون خطا استفاده شود. در صورت عدم وجود چنین تلاشی (یا اگر تلاش موفقیت‌آمیز نباشد) احتمالاً خرابی رخ خواهد داد. دو نوع اساساً متفاوت از تکنیک‌های بازیابی وجود دارد. تکنیک بازیابی رو به عقب شامل کنار گذاشتن حالت فعلی (خراب) به نفع حالت قبلی است (طبیعتاً به مکانیسم‌هایی برای ثبت و ذخیره حالت‌های سیستم نیاز است).

✓درمان اولیه و ادامه سرویس‌دهی سیستم

پس از انجام بازیابی، ضروری است که اطمینان حاصل شود که عملکرد عادی سیستم بدون بروز مجدد خطا ادامه خواهد یافت. اگر تصور شود که خطا گذرا است، هیچ اقدام خاصی لازم نیست، در غیر این صورت، خطا باید از سیستم حذف شود. اولین جنبه درمان خطا، تلاش برای یافتن خطا است. پس از این، می‌توان اقداماتی را برای تعمیر خطا یا پیکربندی مجدد بقیه سیستم برای جلوگیری از خطا انجام داد.

روش شناسایی ساختار بلوک بازیابی به صورت زیر است:

Ensure (acceptance test**) by P0 else-by P1 else** fail;

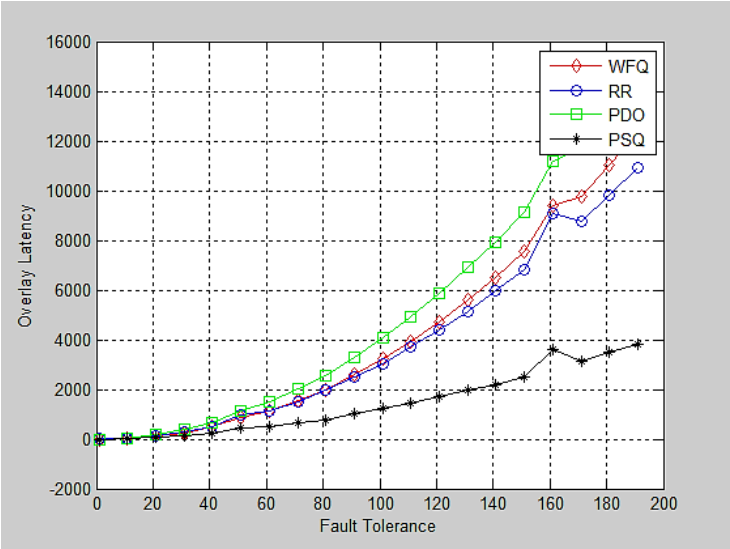
که یک سیستم نرم‌افزاری با چهار مؤلفه را نشان می‌دهد، دو رویه P0 (اولیه) P1 (جایگزین)، آزمون پذیرش و مجموعه‌ای از متغیرهای سراسری که برای رویه‌ها قابل دسترسی هستند. الگوریتم سیستم، ساختار کنترلی است که توسط نحو بیان می‌شود. اگر فرض کنیم که آزمون پذیرش «کامل» است (یعنی تمام نقض‌های مشخصات را تشخیص می‌دهد)، بلوک بازیابی نشان داده شده می‌تواند خطاهایی را در رویه P0 (در صورت وجود) که می‌تواند منجر به شکست آن شود، تحمل کند، البته به شرطی که P1 از آزمون عبور کند. در مورد P0 به عنوان یک سیستم، خطاهای آن اساساً خطاهای طراحی هستند. بنابراین، وقتی گفته می‌شود که «یک بلوک بازیابی می‌تواند خطاهای طراحی را تحمل کند»، منظور واقعی این است که می‌تواند خطاهایی را در برخی از اجزای خود (در مورد ما P0) که می‌توانند به دلیل خطاهای طراحی در آنها شکست بخورند، تحمل کند. در ادامه خواهیم دید که چگونه چهار جنبه تحمل خطا در یک بلوک بازیابی گنجانده شده است. آزمون پذیرش (یک عبارت بولی) برای تشخیص خطاها استفاده می‌شود. از آنجایی که محدوده شناسایی ضریبی بین ۰ تا ۱ است، رویکرد روش های پایه تحقیق در هر ۴ فاز عملیاتی در جدول ۱ آمده است:

جدول ۱ – مقایسه روش های پایه در تحمل پذیری خطا در گام های ۱ تا ۴

PDO	PSQ	WFQ	RR	Fault Tolerance Stage
0.2190	0.3236	0.1211	0.3500	Error Detection
0.1901	0.3212	0.1522	0.3502	Damage Assessment
0.0810	0.3247	0.1497	0.3446	Error Recovery
0.0665	0.3223	0.1498	0.3448	Fault Treatment
94.41	90.47	92.37	91.88	Accuracy

UNCORRECTED PROOF

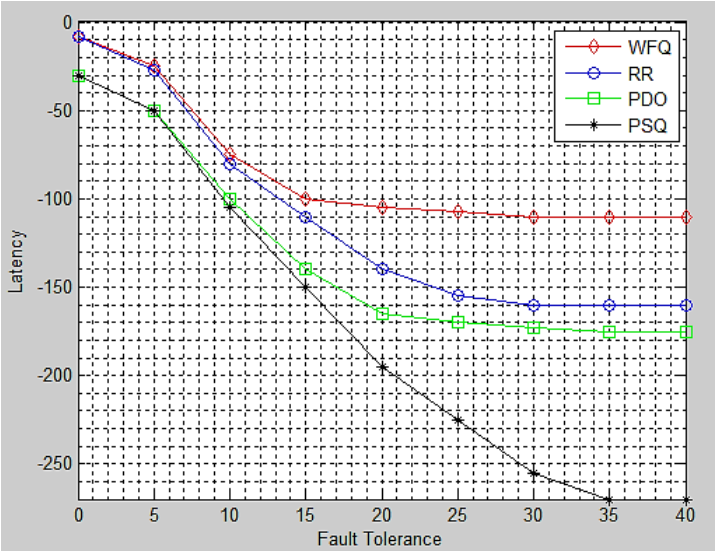
مقادیر آماری نشان دهنده رویکرد تحمل پذیری خطا در روش های پایه است. این میزان هر چه به سمت صفر میل کند، نمایش دهنده اشتیاق سیستم برای تحمل پذیری خطاست، در نتیجه روش PDO نسبت به بقیه بهتر ظاهر شده است.



شکل ۱۵ – نمودار Overlay Latency ۴ روش پایه در تحمل پذیری خطا

شکل ۱۵ میانگین چگونگی تاثیر تحمل پذیری خطا و تاخیر دائمی در مقابل اندازه میانگین تحمل پذیری خطا را نشان می‌دهد. الگوریتم PDO نسبت به بقیه الگوریتم‌ها تحمل پذیری بیشتری را نشان می‌دهد. نتایج WFQ و RR بسیار نزدیک به یکدیگر هستند زیرا تخصیص درخواست‌ها در منابع در هر دوی این الگوریتم‌ها بر اساس اولویت به دست می‌آید. در عین حال روش PSQ پایین ترین تحمل پذیری خطا را نشان می‌دهد و دلیل آن نحوه تخصیص درخواست‌ها به روش PSQ بدون در نظر گرفتن اولویت‌های درخواست است.

UNCORRECTED PROOF



شکل۱۶ – نمودار Latency ۴ روش پایه در تحمل پذیری خطا

شکل ۱۶ میانگین چگونگی تاثیر تحمل پذیری خطا و تاخیر در مقابل اندازه میانگین تحمل پذیری خطا را نشان می‌دهد. الگوریتم PDO بهترین تحمل پذیری خطا را در مقایسه با سایر الگوریتم‌ها نشان می‌دهد. نتایج هر ۴ روش تا تحمل پذیری خطای ۱۵ تقریبا یکسان راست اما با افزایش احتمال تمحل پذیری نتایج RR و PDO بسیار نزدیک به یکدیگر می شوند. با این حال این تفاوت با افزایش میانگین میانگین تاخیرها افزایش می‌یابد. از سوی دیگر اگر میانگین تاخیر به طور متوسط بیشتر از ۱۵۰ بسته باشد. الگوریتم PDO تقریباً تحمل پذیری بهتری دارد. روش‌های WFQ و RR تقریباً از PDO رویکرد کمتری داشته و PSQ از همه ضعیف تر عمل کرده است.

۵-نتیجه‌گیری

در این مقاله مسئله زمان‌بندی درخواست‌های اینترنت اشیا برای به حداقل رساندن تأخیر در محاسبات ترکیبی مه-ابری با استفاده از یک الگوریتم فرا ابتکاری جدید مدل‌سازی شد. تأخیر در این کار به عنوان زمان رفت و برگشت برای پردازش یک درخواست اینترنت اشیا از لحظه شروع آن تا زمانی که به طور کامل پردازش می شود و نتایج به درخواست کننده بازگردانده می شود در نظر گرفته می شود. علاوه بر تأخیر به مصرف انرژی در این مسیر و کمینه‌سازی آن پرداخته شده است. در اینجا برای بهینه‌سازی یک الگوریتم جدید مبتنی بر دفاع گیاهان با نام الگوریتم بهینه‌سازی دفاع گیاهان معرفی شده است. این الگوریتم برگرفته از رفتار ژنتیکی گیاهان برای دفاع می‌باشد. در این الگوریتم در ابتدا یک جامعه تصادفی تولید می شود بر اساس الگوریتم خوشه‌بندی K-means این جمعیت به ژن‌های مختلف خوشه‌بندی می شود هر عضو از این جمعیت نشان دهنده یک جواب یعنی تخصیص درخواست به منبع است و در نهایت عملگرهای معرفی شده در این مقاله بر روی اعضا جامعه اعمال می شود به طور کلی عملگرها بر اساس جستجوی محلی و جستجوی سراسری سازمان‌دهی شده‌اند. هدف از رویکرد محلی تسریع همگرایی و هدف از جستجوی سراسری جلوگیری از قرار گرفتن الگوریتم در بهینه محلی است. تابع هدف در اینجا از دو بخش تشکیل شده است یکی کمینه‌سازی میزان تأخیر و دیگری کمینه‌سازی مصرف انرژی میزان تأخیر سرویس به دست آمده از الگوریتم بهینه‌سازی دفاع گیاهان نیز با سایر تکنیک‌های زمان‌بندی مقایسه شد. نتایج رویکرد پیشنهادی بهبود قابل توجهی را در تأخیر کلی از ۲۳۸۴ درصد و تا ۴۸۵۱ درصد راه حل‌های بهتر نسبت به سایر الگوریتم‌ها نشان داد. نتایج نشان می‌دهد در محاسبات مه به طور کلی نودهای مه تأخیر ارتباطی کمی دارند. این قابلیت استفاده از منابع محاسباتی مه کم مصرف با اندازه کوچک را فراهم می‌کند و تأخیر

UNCORRECTED PROOF

سرویس بهتری را نسبت به استفاده از محاسبات ابری ارائه می‌دهد. همچنین در بررسی تکمیلی تحمل پذیری خطا و تاثیر آن بر تابع هزینه با اعمال تابع هدف، میزان دقت آن در روش PDO با ۹۴.۴۱٪ نسبت به ۳ روش دگیر از استواری قابل توجهی برخوردار است.

در این پژوهش، عملکرد روش پیشنهادی عمدتاً با الگوریتم‌های کلاسیک زمان‌بندی مانند WFQ، PSQ و RR سنجیده شده است. گرچه این امر شناخت مناسبی از بهبودهای به‌دست آمده نسبت به روش‌های پراستفاده فعلی فراهم می‌کند، ولی از مهم‌ترین محدودیت‌های این تحقیق محسوب می‌شود. در سال‌های اخیر، پژوهشگران متعددی از رویکردهای یادگیری ماشین و ترکیب فراابتکاری‌ها یا حتی یادگیری تقویتی عمیق برای حل مساله زمان‌بندی در محیط‌های مه/ابر استفاده کرده‌اند [۱،۲]. این دسته از الگوریتم‌ها عموماً ظرفیت بهبود عملکرد و سازگاری بیشتر با شرایط پویا و مقیاس‌پذیر بالا را دارند، هرچند پیاده‌سازی و تنظیم آن‌ها نیازمند هزینه محاسباتی و زمان آموزش بیشتری است. مقایسه عددی و عملی روش پیشنهادی با الگوریتم‌های مبتنی بر یادگیری عمیق و بهینه‌سازی ترکیبی به عنوان مسیر آینده این پژوهش مدنظر قرار می‌گیرد.

یکی از چالش‌های کلیدی در محیط‌های ترکیبی مه-ابر، وقوع خطا در نودهای لبه و سرورهای مه است، که می‌تواند باعث افزایش تاخیر یا ازدست‌رفتن سرویس‌دهی شود. مدل فعلی با فرض در دسترس بودن همیشگی منابع عمل می‌کند و استراتژی اختصاص وظایف را بر همین اساس نمایش می‌دهد. با این حال، بروز وقفه یا شکست در یک نود مه ممکن است تمامی وظایف اختصاص‌یافته به آن را با شکست مواجه کند و منجر به رد یا تاخیر بیش از حد گردد.

برای افزایش قابلیت اطمینان، می‌توان یک مکانیزم ساده تحمل‌پذیری خطا را چنین به مدل افزود: در صورت شناسایی وقوع خطا یا از دست‌رفتن یک نود، وظایفِ اختصاص یافته به آن بلافاصله به نزدیک‌ترین یا در دسترس‌ترین نود سالم دیگر مجدداً تخصیص داده شوند. بعضی مدل‌ها نیز جهت افزایش محافظت، نسخه پشتیبانی از وظایف را همزمان در بیش از یک نود ذخیره یا اجرا می‌کنند تا در صورت بروز خطا، بدون وقفه طولانی سرویس ادامه یابد [۱،۲]. افزودن چنین سیاست‌هایی به چارچوب زمان‌بندی پیشنهادی، نه‌تنها می‌تواند تأخیر ناشی از خطا را کاهش دهد بلکه پایداری و کیفیت سرویس را به طور معناداری افزایش خواهد داد. طراحی و شبیه‌سازی دقیق رفتار مدل در حضور نرخ‌های خطا به عنوان یک مسیر توسعه مهم برای آینده مطرح است.

۶- کارهای آینده

یکی از موضوعاتی که در این تحقیق به آن پرداخته نشده است حجم داده‌ها می‌باشد که در تابع هدف الگوریتم پیشنهادی مورد ارزیابی قرار نگرفته است. در صورتی‌که یکی از محدودیت‌هایی که در نودهای مه وجود دارد فضای ذخیره‌سازی اطلاعات است بنابراین پیشنهاد می شود در کارهای بعدی به آن پرداخته شود. موضوع دیگری که در این تحقیق به آن پرداخته نشده است مسئله تحمل‌پذیری خطای الگوریتم است. از آنجایی که محیط فرض شده یک محیط بلادرنگ است بنابراین شکست در پردازش هر درخواست از اهمیت بالایی برخوردار است که در این تحقیق این پارامتر مورد ارزیابی قرار نگرفته است لذا پیشنهاد می شود در تحقیقات آینده مورد بررسی قرار گیرد. از طرفی مدل‌های ارائه‌شده در این تحقیق به بررسی و مقایسه تأثیر منابع محاسباتی ابری و مه بر تأخیر سرویس در سیستم‌های ترکیبی مه-ابری پرداخته است. سه پارامتر اصلی شامل نسبت میانگین تأخیر($\delta f/\delta c$) ، نسبت سرعت پردازش (Pf/Pc) و نسبت تعداد منابع (Nf/Nc) برای تحلیل تأثیرات مختلف بر تأخیر سرویس مورد ارزیابی قرار گرفته‌اند. تغییرات در این پارامترها به‌طور مستقیم بر تأخیر سرویس تأثیر می‌گذارند. به‌ویژه، منابع مه با تأخیر کمتر به دلیل نزدیکی به لبه شبکه، در صورتی که از نظر قدرت پردازش ضعیف‌تر باشند، ممکن است در مقایسه با منابع ابری با پردازش قوی‌تر، تأخیر کمتری ارائه ندهند. همچنین، کاهش تعداد منابع در مه یا کاهش سرعت پردازش آن‌ها باعث افزایش قابل‌توجهی در تأخیر سرویس می‌شود. این نتایج نشان می‌دهند که بهینه‌سازی انتخاب منابع، بر اساس ترکیب مناسب این پارامترها، برای دستیابی به حداقل تأخیر سرویس و بهبود عملکرد سیستم‌های ترکیبی ضروری است.

UNCORRECTED PROOF

منابع

1. .Mahmud, R., R. Kotagiri, and R. Buyya, Fog computing: A taxonomy, survey and future directions, in Internet of everything. 2018, Springer. p. 103-130.
2. Mahmoud, M.M., et al., towards energy-aware fog-enabled cloud of things for healthcare. Computers & Electrical Engineering, 2018. 67: p. 58-69.
3. Andriopoulou, F., T. Dagiuklas, and T. Orphanoudakis, Integrating IoT and fog computing for healthcare service delivery, in Components and Services for IoT Platforms. 2017, Springer. p. 213-232.
4. Souza, V.B., et al. Towards distributed service allocation in fog-to-cloud (F2C) scenarios. In Global Communications Conference (GLOBECOM), 2016 IEEE. 2016. IEEE.
5. Bitam, S., S. Zeadally, and A. Mellouk, Fog computing job scheduling optimization based on bees swarm. Enterprise Information Systems, 2018. 12(4): p. 373-397.
6. Wang, J., et al., FESTAL: fault-tolerant elastic scheduling algorithm for real-time tasks in virtualized clouds. IEEE Transactions on Computers, 2014. 64(9): p. 2545-2558.
7. Al-Omari, R, .A.K. Somani, and G. Manimaran, Efficient overloading techniques for primary-backup scheduling in real-time systems. Journal of Parallel and Distributed Computing, 2004. 64(5): p. 629-648.
8. Yin, L., J. Luo, and H. Luo, Tasks scheduling and resource allocation in fog computing based on containers for smart manufacturing. IEEE Transactions on Industrial Informatics, 2018. 14(10): p. 4712-4721.
9. Nguyen, B.M., H. Thi Thanh Binh, and B. Do Son, Evolutionary Algorithms to Optimize Task Scheduling Problem for the IoT Based Bag-of-Tasks Application in Cloud–Fog Computing Environment. Applied Sciences, 2019. 9(9): p. 1730.
10. Li, G., et al., Methods of Resource Scheduling Based on Optimized Fuzzy Clustering in Fog Computing. Sensors, 2019. 19(9): p. 2122.
11. Liu, Q., et al. Task scheduling in fog enabled Internet of Things for smart cities. In 2017 IEEE 17th International Conference on Communication Technology (ICCT). 2017. IEEE.
12. Ghanavati, S., J.H. Abawajy, and D. Izadi, An Energy Aware Task Scheduling Model Using Ant-Mating Optimization in Fog Computing Environment. IEEE Transactions on Services Computing, 2020.
13. Cardellini, V., et al. On QoS-aware scheduling of data stream applications over fog computing infrastructures. In 2015 IEEE Symposium on Computers and Communication (ISCC). 2015. IEEE.
14. Oueis, J., E.C. Strinati, and S. Barbarossa. The fog balancing: Load distribution for small cell cloud computing. In 2015 IEEE 81st vehicular technology conference (VTC spring). 2015. IEEE.
15. De Vries, J ,.et al., Optimal plant defence under competition for light and nutrients: an evolutionary modelling approach. in silico Plants, 2020. 2(1): p. diaa008 .
16. Caraveo, C., F. Valdez, and O. Castillo, A new optimization meta-heuristic algorithm based on self-defense mechanism of the plants with three reproduction operators. Soft Computing, 2018. 22(15): p. 4907-4920.
17. Oma, R., et al., An energy-efficient model for fog computing in the internet of things (IoT). Internet of Things, 2018. 1: p. 14-26 .
18. Vlot, A.C., D.M.A. Dempsey, and D.F. Klessig, Salicylic acid, a multifaceted hormone to combat disease. Annual review of phytopathology, 2009. 47: p. 177-206
19. De Vleesschauwer, D., G. Gheysen, and M. Höfte, Hormone defense networking in rice: tales from a different world. Trends in plant science, 2013. 18(10): p. 555-565.
20. Sugano, S., et al., Role of OsNPR1 in rice defense program as revealed by genome-wide expression analysis. Plant molecular biology, 2010. 74(6): p. 549-562.
21. Peng, X., et al ,.Constitutive expression of rice WRKY30 gene increases the endogenous jasmonic acid accumulation, PR gene expression and resistance to fungal pathogens in rice. Planta, 2012. 236(5): p. 1485-1498.
22. Li, R., et al., OsNPR1 negatively regulates herbivore-induced JA and ethylene signaling and plant resistance to a chewing herbivore in rice. Physiologia plantarum, 2013. 147(3): p. 340-351.
23. Garg, R., A.K. Tyagi, and M. Jain, Microarray analysis reveals overlapping and specific transcriptional responses to different

UNCORRECTED PROOF

plant hormones in rice. Plant signaling & behavior, 2012. 7(8): p. 951-956.
24. Enokido, T., A. Aikebaier, and M. Takizawa. An integrated power consumption model for communication and transaction based applications. in 2011 IEEE International Conference on Advanced Information Networking and Applications. 2011. IEEE.

UNCORRECTED PROOF