



بهینه‌سازی زمان‌بندی در سیستم‌های ابری ناهمگن با الگوریتم‌های کاپوچین و کلونی

مورچگان

صفدر رستمی^{*(۱)}

(۱) گروه مهندسی کامپیوتر، واحد کرمانشاه، دانشگاه آزاد اسلامی، کرمانشاه، ایران*

تاریخ دریافت: ۱۴۰۴/۰۴/۰۱ تاریخ پذیرش: ۱۴۰۴/۰۶/۲۵

چکیده

گسترش سریع مراکز داده ابری، که ناشی از افزایش تقاضا برای خدمات متنوع کاربران است، مصرف انرژی و انتشار گازهای گلخانه‌ای را افزایش داده، است. تقاضا برای زمان‌بندی کارآمد کارها در محاسبات ابری با ظهور محیط‌های ابری پویا و ناهمگن، به طور قابل توجهی افزایش یافته است. الگوریتم‌های زمان‌بندی سنتی، اگرچه در سیستم‌های ساده‌تر مؤثر هستند، اما نمی‌توانند الزامات پیچیده زیرساخت‌های ابری مدرن را برآورده کنند. این محدودیت‌ها، نیاز به راه‌حل‌های مبتنی بر هوش مصنوعی را که سازگاری، مقیاس‌پذیری و بهره‌وری انرژی را ارائه می‌دهند، برمی‌انگیزد. این مقاله یک چارچوب زمان‌بندی وظایف چندهدفه را برای سیستم‌های ابری ناهمگن و آگاه معرفی می‌کند. این چارچوب با ادغام یک مدل پیش‌بینی بار کاری مبتنی بر قوانین فرکتال با الگوریتم‌های فراابتکاری کاپوچین و کلونی مورچگان معکوس، یک رویکرد تطبیقی، پویا و با بهره‌وری انرژی بالا را ارائه می‌دهد. مدل فرکتال نقش حیاتی در پیش‌بینی دقیق بار میزبان‌ها و بهینه‌سازی جریان داده‌ها ایفا می‌کند، که پایه و اساس تصمیم‌گیری‌های زمان‌بندی و مهاجرت ماشین‌های مجازی را تشکیل می‌دهد هدف اصلی این رویکرد، کاهش قابل توجه مصرف انرژی مراکز داده ابری در عین حفظ سطوح بالای استفاده از منابع، بهره‌وری زمان و کیفیت خدمات است. نتایج جامع آزمایش‌ها و شبیه‌سازی‌ها (در محیط نرم‌افزار متلب نسخه ۲۰۱۸، تحت سیستم عامل ویندوز نسخه ۸،۱) نشان می‌دهد که رویکرد پیشنهادی، به لطف قابلیت‌های پیش‌بینی و بهینه‌سازی فرکتال، از نظر انرژی، تعادل بار و زمان اجرا نسبت به روش‌های پیشین ۱۵-۲۰ درصد بهبود داشته است.

کلمات کلیدی: رایانش ابری، زمان‌بندی، الگوریتم جستجوی کاپوچین، مهاجرت ماشین‌های مجازی.

*عهده‌دار مکاتبات:

صفدر رستمی

نشانی: گروه مهندسی کامپیوتر، واحد کرمانشاه، دانشگاه آزاد اسلامی، کرمانشاه، ایران

پست الکترونیکی: safdar_rostami@iau.ac.ir

محاسبات ابری نحوه مدیریت، توزیع و استفاده از منابع محاسباتی را متحول کرده و صنایع را در سراسر جهان متحول ساخته است [1]. رایانش ابری درواقع ترکیبی است از محاسبات مه^۱، محاسبات لبه^۲، محاسبات شبکه^۳، محاسبات همگانی^۴ و محاسبات خودمختار^۵ [2]. یکی از چالش‌های حیاتی که در محیط‌های ابری پدیدار می‌شود، زمان‌بندی کارآمد کارها است. بهینه‌سازی تخصیص وظایف و مدیریت منابع با رشد و تنوع زیر ساخت‌های ابری، اهمیت فزاینده‌ای پیدا می‌کند. زمان‌بندی مؤثر کارها، عملکرد روان سرویس‌های ابری را با متعادل کردن توزیع بار کاری، به حداقل رساندن تأخیرهای پردازش و افزایش عملکرد کلی تضمین می‌کند. زمان‌بندی کارها در محیط‌های محاسبات ابری شامل چندین عامل حیاتی مانند در دسترس بودن منابع، اولویت وظایف، مصرف انرژی و ناهمگونی منابع است [3]. در این راستا، برای برآورده کردن نیازهای مختلف کاربران، انواع مختلفی از ابرها مانند عمومی، خصوصی، ترکیبی، موبایل و فدراسیون‌های ابری^۶ توسعه یافته است [4]. پیچیدگی بالای محیط ابری و الزامات سخت‌گیرانه برای کنترل زمان‌بندی چالش‌های اساسی برای توسعه مدل‌های بهینه‌سازی ایجاد می‌کنند [5]. از آنجایی که میلیون‌ها دستورالعمل در هر ثانیه^۷ و پهنای باند شبکه تحت تأثیر عوامل مختلفی مانند دوره‌های اوج شبکه، استفاده از منابع و تعداد دستگاه‌های شبکه قرار دارند، در فرآیند زمان‌بندی وظایف ابری عدم اطمینان وجود دارد. علاوه بر این، برخی از الگوریتم‌های تکاملی چندهدفه نمی‌توانند برای حل همه مدل‌ها استفاده شوند. بنابراین، یک الگوریتم تکاملی چندهدفه مناسب برای مقابله با مشکل زمان‌بندی وظایف ابری مشخص شده مورد نیاز است. در طول دو دهه گذشته، الگوریتم‌های فراابتکاری^۸ به دلیل کارآمدی آنها در حل مسائل محاسباتی پیچیده و بزرگ مورد توجه بسیاری از محققان قرار گرفته‌اند. الگوریتم‌های فراابتکاری دارای ویژگی‌های مفیدی هستند. (۱) هیچ‌کدام از آنها وابسته به مسئله نیستند، (۲) فضای جستجو به طور مؤثر توسط این الگوریتم‌ها برای یافتن راه‌حلی فرعی یا تقریباً بهینه برای مسائل NP-کامل کاوش می‌شود، و (۳) این الگوریتم‌ها معمولاً غیرقطعی و مبتنی بر تقریب هستند. الگوریتم‌های فراابتکاری به دلیل استقلال غریزی آنها از مسئله قابل حل، برای حل مسائل حوزه‌های مختلف، با عملکرد بسیار قابل قبول، قابل اجرا هستند. استراتژی‌های فراابتکاری معمولاً به عنوان روش‌های مؤثر برای حل مسائل بهینه‌سازی NP-hard با کارایی بالا استفاده می‌شوند. محیط‌های ابری

¹ Fog computing

² Edge computing

³ Grid computing

⁴ Utility computing

⁵ Autonomic computing

⁶ Cloud federations

⁷ millions of instructions per second (MIPS)

⁸ meta-heuristic algorithms

در حال حاضر توسط اکثر الگوریتم‌های فراابتکاری پشتیبانی می‌شوند که برای به دست آوردن راه‌حل‌های نزدیک بهینه (تقریبی) برای مسائل NP-complete در مدت زمان کوتاهی به کار گرفته می‌شوند. در این راستا، فرایند زمان‌بندی کار نسبت به سایر مسائل معمولاً برای یافتن یک راه‌حل بهینه زمان زیادی را می‌طلبد.

در این راستا، مقاله حاضر یک رویکرد نوآورانه را برای زمان‌بندی وظایف در محیط‌های ابری ناهمگن معرفی می‌کند که بر پایه یک مدل پیش‌بینی توانمند مبتنی بر قوانین فرکتال بنا شده است. این مدل فرکتال، با قابلیت خود در شناسایی الگوهای پیچیده و خود متشابه در بارهای کاری ابری، امکان پیش‌بینی دقیق وضعیت منابع را فراهم می‌آورد. این پیش‌بینی دقیق، به عنوان یک لایه هوشمند اولیه، تصمیم‌گیری‌های بعدی الگوریتم‌های فراابتکاری کلونی مورچگان معکوس و کاپوچین را در زمان‌بندی و مهاجرت ماشین‌های مجازی بهینه می‌سازد. بنابراین، تمرکز اصلی ما بر توسعه یک چارچوب جامع است که در آن مدل پیش‌بینی فرکتال نه تنها بینش‌های ارزشمندی در مورد توصیف بار کاری ارائه می‌دهد، بلکه به عنوان یک جزء فعال و حیاتی، کارایی و بهره‌وری انرژی کل سیستم را به طور قابل توجهی افزایش می‌دهد. به طور اختصار اهداف این مقاله علاوه بر زمان‌بندی صحیح کارها شامل پیشنهاد یک مدل بهینه‌سازی زمان‌بندی وظایف ابری چندهدفه با در نظر گرفتن پنج هدف بهینه‌سازی یعنی مصرف انرژی، نرخ خطا، تعادل بار، زمان اجرا و زمان مهاجرت.

ادامه این مقاله به شرح زیر سازماندهی شده است. در بخش دوم کارهای گذشته و مرتبط درخصوص زمان‌بندی وظایف ابری ارائه می‌شود. در بخش سوم مشکل زمان‌بندی وظایف تشریح شده و روش پیشنهادی معرفی می‌شود. در بخش چهارم نتایج حاصل از شبیه‌سازی روش پیشنهادی ارائه شده و عملکرد روش پیشنهادی با روش‌های پیشین مقایسه می‌شود. در نهایت در بخش پنجم نتیجه‌گیری مقاله ارائه می‌شود.

۲- مرور ادبیات

سرویس‌های ابری معمولاً در سه شکل اصلی شامل پلتفرم به عنوان سرویس^۱، زیرساخت به عنوان سرویس^۲ و نرم‌افزار به عنوان سرویس^۳ ارائه می‌شود [6]. انعطاف‌پذیری و مقیاس‌پذیری ارائه شده توسط سرویس‌های ابری، آنها را برای کسب‌وکارها و سازمان‌های سراسر جهان ضروری کرده است. با این حال، با رشد مداوم رایانش ابری، بهینه‌سازی تخصیص و زمان‌بندی کارها به طور فزاینده‌ای پیچیده می‌شود و نیاز به استراتژی‌های پیچیده‌ای برای اطمینان از استفاده

¹ Platform as a Service (PaaS)

² Infrastructure as a Service (IaaS)

³ Software as a Service (SaaS)

و عملکرد کارآمد از منابع دارد [7]. SaaS تسهیلاتی را برای استفاده از برنامه‌های ذخیره‌شده در فضای ابری مانند مدیریت دیتابیس، مدیریت مالی، مدیریت فروش، حسابداری و مدیریت ارتباط با مشتری فراهم می‌کند و به مشتریان اجازه می‌دهد تا آنها را اجرا کنند [8]. IaaS شکلی از محاسبات ابری است که در آن دارایی‌های فیزیکی مانند سرورها، ابزار ذخیره‌سازی، پردازنده، حافظه و زیرساخت‌ها در اختیار کاربران قرار می‌گیرد [9]. در محیط‌های ابری به دلیل حجم زیاد وظایف، ناهمگونی منابع و ماهیت پویای حجم کار، ناکافی است. محیط‌های ابری بسیار پویا هستند و تقاضای منابع می‌تواند به سرعت و غیرقابل پیش‌بینی تغییر کند. این نوسانات، عملکرد بهینه الگوریتم‌های زمان‌بندی ایستا را دشوار می‌کند. در نتیجه، نیاز به رویکردهای زمان‌بندی تطبیقی آشکار شده است [10]. در طول سال‌های گذشته، مصرف بالای انرژی توسط مراکز داده ابری به یک چالش اساسی تبدیل شده است. به‌طور کلی، انرژی این تجهیزات به دلایل متعددی مانند ناکارآمدی در سیستم خنک‌کننده مرکز داده و به‌کارگیری تجهیزات شبکه و سرورها هدر می‌رود [11].

برای مدیریت تصمیم‌گیری‌های پیچیده در زمان واقعی الگوریتم‌های زمان‌بندی کلاسیک عمدتاً بر قوانین از پیش تعریف شده یا اولویت‌های ایستا متکی بوده‌اند که احتمالاً در ماهیت سیال محاسبات ابری مناسب نیستند [12]. الگوریتم‌های هوش مصنوعی ممکن است ویژگی‌های وظایف را همراه با در دسترس بودن منابع و معیارهای عملکرد سیستم تجزیه و تحلیل کنند تا تصمیمات زمان‌بندی بهتر و آگاهانه‌تری را ممکن سازند یکی دیگر از چالش‌های قابل توجه در زمان‌بندی کار مبتنی برابر، ناهمگونی در منابع است [13]. معمولاً یک محیط ابری شامل منابع مختلفی با قابلیت‌های پردازش، ظرفیت‌های ذخیره‌سازی و پروفایل‌های مصرف انرژی متغیر است. الگوریتم‌های زمان‌بندی مبتنی بر هوش مصنوعی می‌توانند ناهمگونی در منابع را در نظر بگیرند و تخصیص منابع را بر اساس این تفاوت‌ها بهینه کنند. به عنوان مثال، الگوریتم‌های مبتنی بر هوش مصنوعی می‌توانند عملکرد منابع مختلف را برای وظایف خاص پیش‌بینی کنند تا اطمینان حاصل شود که کارها به مناسب‌ترین منابع اختصاص داده می‌شوند. مراکز داده‌ای که زیرساخت‌های ابری را تغذیه می‌کنند، مقدار زیادی انرژی مصرف می‌کنند. زمان‌بندی کار مبتنی بر هوش مصنوعی می‌تواند با بهینه‌سازی رزرو منابع و در نتیجه به حداقل رساندن اتلاف ناشی از بیکاری آنها، به کاهش مصرف انرژی کمک کند [14]. الگوریتم‌های هوش مصنوعی می‌توانند هوشمندانه کارها را بر اساس اهمیت، فوریت و منابع مورد نیازشان اولویت‌بندی کنند تا اطمینان حاصل شود که وظایف با اولویت بالا در بازه‌های زمانی مورد نیاز تکمیل می‌شوند و در عین حال عملکرد کلی سیستم را حفظ می‌کنند. ادغام هوش مصنوعی در عملیات مراکز داده، به طور قابل توجهی بهره‌وری انرژی را افزایش داده، انتشار کربن را کاهش و با اهداف محاسبات ابری سبز همسو است. این چارچوب از یادگیری تقویتی چندعاملی برای بهینه‌سازی توزیع بار کاری، سیستم‌ها و عملیات باتری در زمان واقعی استفاده می‌کند [15].

مشکلات زمان‌بندی وظایف از جنبه‌های مختلفی در محیط ابری بررسی شده‌اند. به‌منظور کاهش مصرف انرژی در مراکز داده، نویسندگان در [16]، دستاوردهای کلیدی شامل انتقال بارهای کاری مقاوم در برابر تأخیر به دوره‌هایی با شدت کربن کمتر، تنظیم پویای سیستم‌های خنک‌کننده برای به حداقل رساندن مصرف انرژی و مدیریت استراتژیک مصرف باتری برای به حداکثر رساندن اتکا به انرژی تجدیدپذیر است. چارچوب DC-CFR از روش‌های سنتی مانند ASHRAE بهتر عمل می‌کند و به کاهش ۱۴,۴۶ درصدی در انتشار کربن، ۱۴,۳۵ درصدی در مصرف انرژی و ۱۳,۶۹ درصدی در هزینه‌های انرژی در طول یک سال در مناطق مختلف دست یافته است. سازگاری و مقیاس‌پذیری بلادرنگ آن، آن را برای محیط‌های ابری پویا و متغیر عملی می‌کند و پتانسیل آن را برای پذیرش گسترده برجسته می‌سازد. این چارچوب با ترکیب استراتژی‌های هوشمند مبتنی بر هوش مصنوعی، مسیری متحول به سوی شیوه‌های محاسبات ابری پایدار و با بهره‌وری انرژی بالا را نشان می‌دهد. در مقاله [17]، برای کاهش مصرف تجهیزات تحت محدودیت زمان تکمیل، از Wi-Fi برای انتقال وظایف بارگذاری شده به یک مرکز ابر استفاده شده است. رویکردهای بسیاری برای رسیدگی به مشکلات استفاده بهینه از منابع و عدم تعادل بار در طول زمان‌بندی وظایف پیشنهاد شده است. نویسندگان در مقاله [18]، یک مدل زمان‌بندی چندهدفه زمان‌بندی وظایف پیشنهاد کرده‌اند که در آن هزینه تکمیل کار، کوتاه‌ترین زمان انتظار و درجه تعادل بار به‌عنوان اهداف بهینه‌سازی در نظر گرفته شده است. برای اطمینان از کیفیت خدمات هنگام بهبود نرخ استفاده از منابع، یک چارچوب آنلاین برای زمان‌بندی وظایف در مراکز داده ابری در مقیاس بزرگ در [19] پیشنهاد شده است. نویسندگان در [20]، از الگوریتم بهینه‌سازی نهنگ^۱ برای زمان‌بندی وظایف ابری با اهداف چندگانه استفاده کرده‌اند. در [21] با ترکیب یادگیری Q با شبکه‌های عصبی عمیق و استفاده از تعادل بار و حداکثر زمان تکمیل کار به‌عنوان شاخص‌های کارایی رویکرد جدیدی را برای حل مشکل زمان‌بندی وظایف در محیط ابری پیشنهاد کرده‌اند. در [22] یک الگوریتم زمان‌بندی اکتشافی مبتنی بر تحرک ادراکی باهدف کاهش زمان و هزینه تکمیل کار ارائه شده است. به‌منظور کاهش سربار ارتباط بین ماشین‌های مجازی، در [23] از CloudSim برای تحقق زمان‌بندی وظایف مستقل استفاده شده است. در رویکردهای زمان‌بندی پیشنهاد شده در [24, 25] زمان و هزینه زمان‌بندی وظایف به‌طور قابل توجهی کاهش یافته است. با توجه به مطالعات انجام یافته زمان‌بندی وظایف ابری یک مسئله بهینه‌سازی چندهدفه^۲ است که در آن بسیاری از اهداف بهینه‌سازی باید در نظر گرفته شوند. علاوه بر این، نویسندگان در [26]، با هدف حل مشکل تراکم شبکه و اتلاف پهنای باند یک الگوریتم درک مسیر مشترک را طراحی کرده‌اند. همچنین، برای حل مشکل قابلیت اطمینان ناشی از شکست منابع ابری، یک الگوریتم زمان‌بندی تحمل‌پذیر خطا و بلادرنگ در [27] پیشنهاد شده

¹ whale optimization algorithm

² Multi-objective optimization problem (MOP)

است. نویسندگان در [28]، اخیراً، زمان‌بندی وظایف خودکار در میان کاربران محاسبات ابری با ظهور Kubernetes، مشهورترین زمان‌بند وظایف، تغییر عمده‌ای را تجربه کرده است که در عوض، هماهنگی استقرارهای بزرگ را ممکن می‌سازد. فناوری‌های DRL ابداعاتی هستند که می‌توانند به نظارت پویا، پیش‌بینی پارامترهای ابری و زمان‌بندی وظایف به صورت بلادرنگ منجر شوند و در نتیجه، استفاده از منابع سیستم و کارایی اجرای وظایف را بهینه کنند. در سال‌های اخیر، الگوریتم‌های تکاملی چندهدفه مختلفی به‌طور گسترده برای حل مسائل بهینه‌سازی چندهدفه مورد استفاده قرار گرفته‌اند. در [29]، از الگوریتم بهینه‌سازی ازدحام ذرات بهبودیافته برای حل مسئله بهینه‌سازی چندهدفه استفاده شده است. در [30]، از یک الگوریتم تکاملی چندهدفه افزایش امنیت اینترنت اشیا صنعتی بلاک چین استفاده شده است. در [31]، الگوریتم NSGA-III براساس استراتژی رتبه‌بندی حداکثر برای حل مسائل تست استاندارد DTLZ پیشنهاد شده است. با این حال، با افزایش تعداد اهداف، الگوریتم‌های تکاملی چندهدفه کاستی‌های مختلفی را از خود نشان می‌دهند مانند نگهداری ناکافی تنوع، عدم فشار انتخاب و افزایش تضاد بین همگرایی و تنوع [32]. برای غلبه بر این مشکلات، محققان الگوریتم‌های تکاملی چندهدفه را از دو جنبه اصلی یعنی انتخاب محیطی¹ و انتخاب تطبیقی² بهبود بخشیده‌اند [33]. روش‌های انتخاب تطبیقی، فرزندان مناسبی را با انتخاب والدین باکیفیت بالا تولید می‌کنند، مانند تسلط پارتو³، تسلط شبکه⁴ و انتخاب مسابقه‌ای⁵ [35,34]. روش‌های انتخاب محیطی، نسل بعدی راه‌حل‌های برجسته را از راه‌حل‌های فعلی والدین و فرزندان بر اساس معیارهای مشخص انتخاب می‌کنند. به‌طور معمول دو نوع استراتژی در زمان‌بندی انجام کاری در راستای بهینه‌سازی مدیریت بار کاری وجود دارد، این دو روش عبارت‌اند از استراتژی ایستا و پویا. استراتژی ایستا به معنای تدوین یک استراتژی بهینه‌سازی مدیریت کاربری بوده که پس از تدوین و به‌کارگیری آن، دیگر امکان تغییر در آن وجود ندارد؛ اما در استراتژی پویا، حتی پس از تدوین و به‌کارگیری آن، امکان تغییر در استراتژی موردنظر وجود داشته و این تغییرات در برابر با چالش‌هایی است که در زمان استفاده از استراتژی مورد در زمان‌بندی به‌منظور بهینه‌سازی مدیریت بار کاری در زمان حجم بالای درخواست‌ها می‌باشد. در زمان‌بندی پویا، تعداد وظایف، مکان ماشین‌های مجازی و شیوه تخصیص منبع ثابت نیست و دفعات ورود وظایف قبل از ارسال آن‌ها مشخص نیست. زمان‌بندی پویا، وظایف را بر روی منابع در زمان اجرا زمان‌بندی می‌کند تا به یک تعدیل بار میان المان‌های پردازشی دست یابد [36]. علاوه بر الگوریتم‌های فراابتکاری و هوش مصنوعی، رویکردهای نوین دیگری نیز برای مقابله با پیچیدگی‌های محیط‌های ابری مطرح شده‌اند. در سال‌های اخیر، کاربرد مفاهیم هندسه فرکتال و تحلیل

¹ environmental selection

² matching selection

³ Pareto-dominance

⁴ grid dominance

⁵ tournament selection

سری‌های زمانی فرکتالی در مدل‌سازی و پیش‌بینی بارهای کاری پویا و غیرخطی در سیستم‌های توزیع‌شده، از جمله رایانش ابری، توجه فزاینده‌ای را به خود جلب کرده است. مطالعاتی مانند [37,39] نشان داده‌اند که بارهای کاری شبکه و سرورها اغلب دارای ویژگی‌های خودمتشابهی و وابستگی بلندمدت هستند که با مدل‌های سنتی به خوبی قابل مدل‌سازی نیستند. این ویژگی‌های فرکتالی، فرصتی را برای توسعه مدل‌های پیش‌بینی دقیق‌تر فراهم می‌آورد که می‌تواند به تصمیم‌گیری‌های بهینه‌تر در تخصیص منابع و زمان‌بندی وظایف منجر شود. با این حال، یکپارچه‌سازی جامع مدل‌های پیش‌بینی مبتنی بر فرکتال با الگوریتم‌های هوش جمعی برای دستیابی به یک چارچوب زمان‌بندی چندهدفه و آگاه از انرژی، همچنان یک شکاف پژوهشی مهم محسوب می‌شود. مقاله حاضر با هدف پر کردن این شکاف، به ارائه یک رویکرد ترکیبی می‌پردازد که از مزایای هر دو حوزه بهره می‌برد.

۳- روش پیشنهادی

در این مقاله تلاش شده است که یک مدل پیش‌بینی توانمند و معقول بر اساس قوانین فرکتال^۱ و الگوریتم‌های هوش جمعی پیشنهاد شود که این روش می‌تواند در بهینه‌سازی روند زمان‌بندی و بهینه‌سازی مصرف انرژی مراکز ابری مفید واقع شود. پیش‌بینی فرکتال یک حوزه تخصصی در روش‌های پیش‌بینی است که با الگوهای داده‌های پیچیده و خودمتشابه سروکار دارد. این روش از هندسه فرکتال الهام می‌گیرد که ساختارهای هندسی پیچیده و تکرار شونده را در طبیعت و ریاضیات بررسی می‌کند. در زمینه تحقیق ما، "پیش‌بینی فرکتال" در ثبت و مدل‌سازی پیچیدگی ذاتی و خودمتشابهی که اغلب در بارهای کاری محاسبات ابری یافت می‌شود، به ویژه در سناریوهایی که ویژگی‌های متنوع و غیرقابل پیش‌بینی وظایف رایج است، نقش دارد. در حالی که تمرکز اصلی ما بر توسعه یک روش زمان‌بندی وظایف پیشرفته با استفاده از الگوریتم‌های کلونی مورچگان معکوس و کاپوچین است، مفهوم پیش‌بینی فرکتال بینش‌های ارزشمندی در مورد توصیف بار کاری ارائه می‌دهد. در یک محیط محاسبات ابری، درک ماهیت فرکتال بارهای کاری می‌تواند به پیش‌بینی و بهینه‌سازی تخصیص منابع کمک کند، که برای زمان‌بندی کارآمد وظایف ضروری است در این مقاله، نقش محوری قوانین فرکتال در ایجاد یک مدل پیش‌بینی بار کاری دقیق و پویا، به عنوان سنگ بنای روش پیشنهادی ما، مورد تأکید قرار می‌گیرد. قوانین فرکتال، با بهره‌گیری از قابلیت‌های پیش‌بینی سری‌های زمانی، امکان پیش‌بینی بار میزبان‌های فیزیکی را فراهم می‌آورند. این پیش‌بینی فرکتال، که از هندسه فرکتال و الگوهای خودمتشابه در داده‌ها الهام گرفته شده است، به ما اجازه می‌دهد تا پیچیدگی ذاتی و غیرقابل پیش‌بینی بارهای کاری در محیط‌های

¹ Fractal

ابری را مدل‌سازی کنیم. این مدل پیش‌بینی، به عنوان یک لایه هوشمند اولیه، اطلاعات حیاتی را برای الگوریتم‌های هوش جمعی (کلونی مورچگان معکوس و کاپوچین) فراهم می‌کند. به این ترتیب، مدل پیش‌بینی فرکتال نه تنها پیش‌بینی‌های ارزشمندی در مورد توصیف بار کاری ارائه می‌دهد، بلکه به طور فعال در تصمیم‌گیری برای راه‌اندازی زمان‌بند بر اساس پیش‌بینی بار، و همچنین در بهینه‌سازی تخصیص منابع و کاهش مصرف انرژی با حفظ کیفیت سرویس، نقش ایفا می‌کند. بنابراین، روش پیشنهادی ما یک چارچوب یکپارچه است که شامل یک مدل پیش‌بینی مبتنی بر قوانین ریاضی فرکتال و مکانیزم‌های زمان‌بندی و مهاجرت ماشین‌های مجازی بر اساس الگوریتم‌های هوش جمعی است. این هم‌افزایی، کارایی و پایداری سیستم را به طور قابل توجهی افزایش می‌دهد.

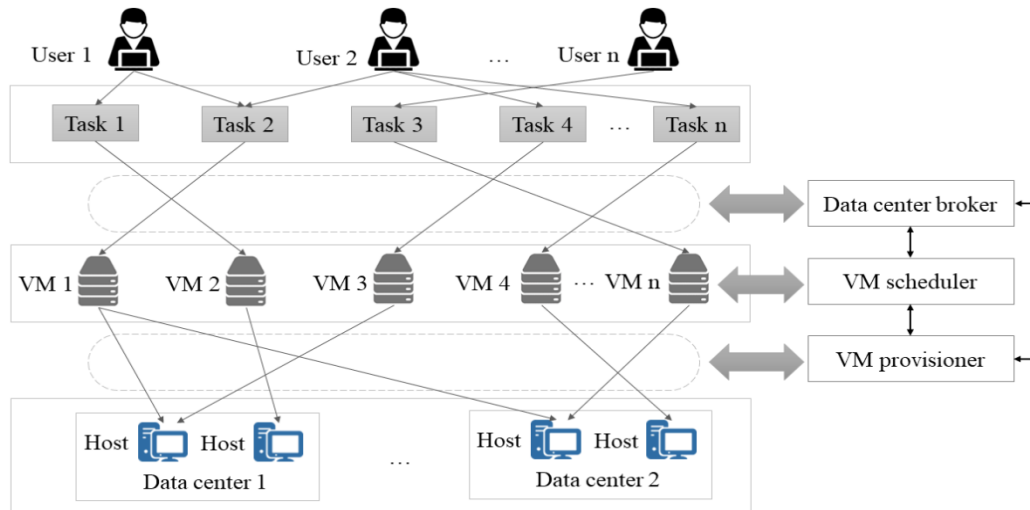
۳-۱- بیان مسئله

در رایانش ابری، فرآیند زمان‌بندی وظایف را می‌توان به دو لایه تقسیم کرد یک لایه مجازی و یک لایه میزبان. لایه مجازی وظایف ابری را به ماشین‌های مجازی مناسب اختصاص می‌دهد و ماشین مجازی انتخاب شده ظرفیت پردازش ثابتی را به هر وظیفه اختصاص می‌دهد. وظیفه لایه میزبان انتخاب میزبان مناسب و استفاده از ماشین مجازی روی آن برای انجام وظایف است. در این مقاله بر روی مشکل زمان‌بندی سطح اول، یعنی تخصیص کارها به ماشین‌های مجازی تمرکز شده است. همان‌طور که در شکل ۱ نشان داده شده است، فرآیند زمان‌بندی وظایف ابری را می‌توان به چهار سطح تقسیم کرد: یک لایه ماشین مجازی، یک لایه کاربر ابری، یک لایه مرکز داده و یک لایه وظیفه. این چهار سطح سه نوع نگاهت را ایجاد می‌کنند: نگاهت یک به چند بین کاربران و وظایف، نگاهت چند به چند بین ماشین‌های مجازی و مراکز داده، و نگاهت چند به چند بین وظایف و ماشین‌های مجازی. کارگزار^۱ مرکز داده مسئول کپسوله‌سازی عملیات و مدیریت ماشین‌های مجازی است. هنگامی که یک کاربر یک درخواست کار ارائه می‌کند، کارگزار مرکز داده محدودیت‌های تقاضای کاربر را تجزیه و تحلیل و پردازش می‌کند. درخواست وظایف کاربران برای اجرا به لایه ماشین مجازی ارسال می‌شود.

با رشد سریع تعداد کاربران ابری، مقیاس زمان‌بندی وظایف به‌شدت افزایش می‌یابد و سرعت تکمیل وظایف ابری به طور اجتناب‌ناپذیر کاهش می‌یابد. اگر بار یک ماشین مجازی در طول فرآیند زمان‌بندی متعادل نباشد، زمان زمان‌بندی وظایف ابری و هزینه آن افزایش می‌یابد؛ بنابراین، هزینه زمان‌بندی، زمان انجام زمان‌بندی، تعادل بار ماشین‌های مجازی و نرخ تکمیل کار ارتباط نزدیکی با هم دارند. برای زمان‌بندی وظایف ابری، لازم است یک طرح تخصیص کار مناسب تدوین شود که نیازهای کاربران را تا حد ممکن برآورده کند. علاوه بر این، تغییرات در تجهیزات شبکه، استفاده از منابع

^۱ Broker

و ساعات اوج مصرف شبکه می‌تواند منجر به نامشخص بودن پهنای باند شبکه و MIPS در طول فرایند زمان‌بندی شود؛ بنابراین، محدوده پهنای باند شبکه و تغییرات MIPS باید توسط کران‌های بالا و پایین مناسب محدود شوند. پهنای باند شبکه و MIPS مستقیماً بر زمان اجرای یک کاربر، هزینه زمان‌بندی و تکمیل کار تأثیر می‌گذارند.



شکل ۱: دیاگرام فرایند زمان‌بندی وظایف ابری

۲-۳- مدل‌های بهینه‌سازی

۳-۲-۱- مدل مصرف انرژی

مدل مصرف انرژی و زمان انجام حجم کاری در رابطه‌های ۱ و ۲ تشریح گردیده است. برای گره i بهره‌وری پردازنده (RPU_i) و بهره‌وری حافظه (RMU_i) را می‌توان به صورت رابطه ۱ و ۲ بیان نمود؛ که در آن n تعداد ماشین‌های مجازی در حال اجرا در گره i ام است و l تعداد کارهای اختصاص یافته به n تا VM است. rpu_{ijk} و rmu_{ijk} بهره‌وری حافظه و پردازنده k کار در حال اجرا در j ماشین مجازی در گره i ام می‌باشد.

$$RPU_i = \sum_{j=1}^v \sum_{k=1}^l rpu_{ijk} \quad (1)$$

$$RMU_i = \sum_{j=1}^v \sum_{k=1}^l rmu_{ijk} \quad (2)$$

گره i ام از لحاظ مصرف پردازنده و حافظه مربوطه، می‌تواند به صورت رابطه ۳ بیان شود که در آن C تعداد واحدهای حافظه است و t واحد زمان می‌باشد؛ بنابراین، مصرف انرژی باید نرمال شود برای اینکه در تابع هدف استفاده خواهد شد. مصرف برق نرمال شده را می‌توان به صورت رابطه ۴ نشان داد که در آن $EC_{\min}$ و $EC_{\max}$ حداکثر و حداقل برق مصرفی سیستم می‌باشد. همچنین مصرف انرژی کل سیستم مطابق رابطه ۵ به دست می‌آید که در آن M تعداد کل گره‌ها است.

$$EC_i = \frac{(RPU_i)(RMU_i)}{C} \times t \quad (3)$$

$$N(EC_i) = \frac{EC_i - EC_{\min}}{EC_{\max} - EC_{\min}} \quad (4)$$

$$EC = \sum_{i=1}^M EC_i \quad (5)$$

۳-۲-۲- مدل زمان اجرای کارها

برای محاسبه زمان اجرای کارها از رابطه ۶ استفاده می‌شود که در آن ET_i کارهای در حال اجرا در ماشین‌های مجازی در گره i و ET_{ijk} زمان اجرای کارهای K کار در حال اجرا در j ماشین مجازی در گره i ام می‌باشند. همچنین برای محاسبه زمان اجرای کارها برای کل سیستم از رابطه ۷ استفاده می‌شود.

$$ET_i = \sum_{j=1}^v \sum_{k=1}^l ET_{ijk} \quad (6)$$

$$ET = \sum_{i=1}^M ET_i \quad (7)$$

۳-۲-۳- تابع هدف

تابع هدف به عنوان یکی از عوامل کلیدی برای هر الگوریتم بهینه‌سازی در نظر گرفته می‌شود. با توجه به اهداف در نظر گرفته شده در این مقاله یعنی کاهش انرژی مصرفی و زمان اجرای کارها، با یک مسئله بهینه‌سازی چندهدفه روبه‌رو هستیم. از این‌رو برای رسیدن به یک نتیجه مطلوب بدیهی است همه این شرایط باید به طور هم‌زمان مورد توجه قرار

گیرند. هدف اصلی در این روش پیشنهادی به حداقل رساندن مصرف انرژی (EC) و زمان اجرا (ET) جهت بهره‌وری مؤثر از منابع گره‌ها می‌باشد؛ بنابراین، مقدار برازندگی با استفاده از رابطه ۸ محاسبه می‌شود که در آن w_1 و w_2 ضرایب وزنی بوده و عددی بین صفر و یک می‌باشند و برای نرمال‌سازی مقادیر تابع fit استفاده می‌شود.

$$fit_m = w_1 \times ET + w_2 \times EC \quad (8)$$

تابع هدف در معرض محدودیت‌های رابطه‌های شماره ۹ و ۱۰ قرار می‌گیرد. طبق رابطه ۹ نسبت ظرفیت پردازنده جمع شده ماشین‌های مجازی بر روی گره i به ظرفیت CPU گره i نباید از مقدار آستانه U CPU ام از گره i ام فراتر رود.

$$\frac{\sum_{j=1}^n v c_j}{c_i} \leq cpu_{th} \quad (9)$$

همچنین بر اساس رابطه ۱۰ نسبت حافظه جمع شده ماشین‌های مجازی بر روی گره i به ظرفیت حافظه گره i نباید از مقدار آستانه، MEM از گره i ام فراتر رود.

$$\frac{\sum_{j=1}^n v m_j}{m_i} \leq MEM_{th} \quad (10)$$

همچنین رابطه ۱۱ بیان می‌نماید که کل EC از گره i نباید بیش از مقدار آستانه انرژی آن، ENERGYth فراتر رود.

$$EC_i \leq Energy_{th} \quad (11)$$

۳-۳- گام‌های روش پیشنهادی

۳-۳-۱- جایگذاری ماشین‌های مجازی

در این مرحله تلاش می‌شود برای هر ماشین مجازی یک میزبان مناسب انتخاب شود به گونه‌ای که مجموع منابع مورد نیاز ماشین‌های مجازی که به یک میزبان تخصیص داده شده از منابع در اختیار آن میزبان فراتر نبوده و همچنین

از مقدار کمینه‌ای میزبان‌ها استفاده شود. در این بخش از حالت بهبود یافته الگوریتم اول بهترین کاهشی^۱ که مبتنی بر ظرفیت پردازنده ماشین مجازی است استفاده شده است. این الگوریتم، تمامی ماشین‌های مجازی را براساس میزان بهره‌وری پردازنده آنها به صورت نزولی مرتب می‌کند و هر ماشین مجازی را به میزبانی تخصیص می‌دهد که کم‌ترین افزایش توان مصرفی را به ازای آن تخصیص داشته باشد.

۳-۳-۲- تقسیم‌بندی وظایف براساس الگوریتم بهینه‌سازی کلونی مورچگان معکوس^۲

در این مرحله تقسیم وظایف به استخرها بر اساس میزان بار وظایف (شامل سایز داده‌ها، و ... و فاصله آنها به منابع صورت می‌گیرد. این احتمال تخصیص بر اساس الگوریتم مورچگان معکوس عمل می‌کند. کلیت این روش در ادامه آورده شده است. مهم‌ترین عملیات در فرایند مورچگان معکوس، مقداردهی اولیه فرومون، انتخاب ماشین مجازی برای وظیفه بعدی و به‌روزرسانی فرومون است.

مقداردهی اولیه فرومون: مقدار اولیه فرومون را به صورت دنباله $\tau_{ij}(t)$ برای هر لبه (اتصال) بین وظیفه i و ماشین j تعریف می‌کنیم. مقدار فرومون را برای بار اول ($t=0$) برای همه یال‌ها، برابر و به صورت یک عدد ثابت کوچک τ_0 منظور می‌کنیم.

الگوی انتخاب ماشین مجازی برای وظایف تقسیم شده بعدی: در هر تکرار از الگوریتم مورچگان، هر مورچه ($k=1 \dots m$) که m تعداد مورچه‌ها را نشان می‌دهد یک دوره از n اجرا (n تعداد وظایف) را بر اساس الگوی انتقال احتمالاتی مربوط به خود، تولید می‌کند. به عنوان مثال مورچه k ام ماشین j را براساس رابطه احتمالاتی ۱۲ برای واگذاری کار بعدی انتخاب می‌کند. هر ماشینی که مقدار احتمالاتی آن بیشترین مقدار باشد برای مرحله بعدی انتخاب می‌شود.

$$P_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^k(t)^\alpha \left(\frac{1}{d_{ij}^k}\right)^\beta}{\sum_{s \in allowed_k} \tau_{ij}^s(t)^\alpha \left(\frac{1}{d_{ij}^s}\right)^\beta}, & j \in allowed_k \\ 0, & otherwise \end{cases} \quad (12)$$

در رابطه ۱۲، $\tau_{ij}(t)$ نشان‌دهنده فرومون مربوطه به یال یا لبه اتصالی وظیفه i به ماشین مجازی j در لحظه t است، d_{ij}^k بیان‌کننده فاصله مسیر (i,j) است که توسط مورچه k طی شده است و $allowed_k = \{0, 1, \dots, n-1\}$ بیان‌کننده

^۱ Power Aware Best Fit Decreasing

^۲ Inverted ant colony optimization

ماشین‌های مجازی در نظر گرفته شده برای مورچه k در گام بعدی؛ بنابراین با کسر آن‌ها تنها کاندیداهایی برای انتخاب بعد باقی می‌مانند که قبلاً انتخاب نشده‌اند. در این پیاده‌سازی، زمان مورد انتظار اجرا و زمان انتقال وظیفه i به ماشین مجازی j است که به صورت زیر محاسبه می‌شود.

$$d_{ij} = \frac{TL_Task_i}{Pe_num_j * Pe_mips_j * VM_j} + \frac{InputFileSize}{VM_bw_j} \quad (13)$$

که در آن TL_Task_i طول نهایی وظیفه برای واگذاری به ماشین j ، Pe_num_j تعداد پردازنده‌های ماشین مجازی j ، Pe_mips_j مربوط به MIPS (کارایی) هر یک از پردازنده‌های ماشین مجازی j ، $InputFileSize$ طول وظیفه قبل از اجرا و VM_bw_j پهنای باند ارتباطی ماشین مجازی j است.

پارامترهای α و β شاخص‌های فرا اکتشافی به ترتیب به معنی، اهمیت مسیر یا یال مربوطه با فرومون آن و شاخصی برای عکس زمان مورد انتظار اجرا به معنی تأثیر اطلاعات به دست آمده از روش فرا اکتشافی است. به روزرسانی فرومون: بعد از تکمیل هر دوره، هر مورچه فرومونی را بر اساس رابطه ۱۴ تولید و ردگذاری می‌کند (به هر لبه مقداری را اختصاص می‌دهد).

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k}, & \text{if ant } k \text{ passes } (i,j) \text{ at current iteration} \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

در رابطه ۱۴، L^k نشان‌دهنده طول کل زمان اجرای به دست آمده توسط مورچه در پایان تور (دوره) k ام آن (مقدار مورد انتظار زمان اجرای دوره مربوطه) که توسط رابطه ۱۵ محاسبه می‌شود و Q پارامتری وفقی (سازگاری) است.

$$L^k = \arg \max_{j \in J} \{sum_{i \in IJ} (d_{ij})\} \quad (15)$$

در رابطه ۱۵، IJ مجموعه‌ای از وظایف است که به ماشین j اختصاص داده شده است. بعد از هر تکرار، فرومون مربوط به هر لبه توسط رابطه ۱۶ به روزرسانی می‌شود که در آن ρ پارامتر تبخیر یا میرایی دنباله فرومون است و مقدار آن بین صفر و یک است و مقدار $\Delta\tau_{ij}(t)$ توسط رابطه ۱۷ محاسبه می‌شود که در آن m تعداد مورچه‌ها و k اندیس دوره است.

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (16)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (17)$$

زمانی که همه مورچه‌ها یک پیمایش را کامل کردند، یک دوره از بین دوره‌ها به‌عنوان بهترین دوره بر اساس مقدار فرومون انتخاب می‌شود و با T^+ نام‌گذاری می‌شود و کیفیت آن با Q/L^+ بیان می‌شود که در آن L^+ طول بهترین دوره (T^+) است. این مقدار فرومون که مسیر مناسب و راه‌حل را نشان می‌دهد توسط رابطه ۱۸ تقویت می‌شود تا به‌عنوان زمان‌بندی بهینه مطرح شود.

$$\tau_{ij}(t) = \tau_{ij}(t) + \frac{Q}{L^+} \quad \text{if } (i,j) \in T^+ \quad (18)$$

در روش پیشنهادی، در این مرحله در ابتدا گرافی برای همه وظایف تقسیم‌شده و ماشین‌ها رسم می‌شود و برای واگذاری از هر وظیفه به هر ماشین یالی رسم می‌شود. سپس تعدادی مورچه به‌عنوان رهیاب (یابنده بهترین راه‌حل یا مناسب‌ترین ماشین برای واگذاری کارها) به‌صورت تصادفی در موقعیت‌های مختلف تولید می‌شود. برای هر یال یا لبه که معادل واگذاری کار به ماشین است یک مقدار ثابت به‌عنوان فرومون تولید می‌شود. در مرحله بعد هر یک از این مورچه‌ها بر اساس مقدار احتمالاتی تولید شده (رابطه ۱۲) یکی از ماشین‌ها را که بیشترین مقدار را آورده برای واگذاری کار بعدی از بین ماشین‌هایی که هنوز به آن‌ها کاری واگذار نشده انتخاب می‌کنند. در گام بعد، مقدار فرومون برای واگذاری مربوطه یا همان ارتباط یا یال بر اساس اطلاعات پردازنده و مشخصات لینک ارتباطی با استفاده از روابط ۱۴ و ۱۵ محاسبه می‌شود و فرومون ارتباط مذکور به‌روزرسانی می‌شود. در نهایت دوره هر مسیری که فرومون قوی‌تری دارد با توجه به رابطه ۱۸ به‌عنوان مسیر بهینه انتخاب و نقشه یا گراف واگذاری وظایف به ماشین‌های مجازی مشخص می‌شود.

۳-۳-۳- شناسایی ماشین‌های فیزیکی در حال سرریز شدن

در این مرحله میزبان‌های فیزیکی که در خطر سرریز شدن هستند، شناسایی می‌شوند. نکته مهم در این مرحله این است که پیش از وقوع سرریزی این تشخیص و تصمیم گرفته شود. چرا که وقوع سرریزی مسئله خوشایندی نیست. بنابراین نکته مهم این است که چه زمانی یک میزبان را پریار بدانیم چرا که اگر این مسئله خیلی زود اعلام شود مانع از استفاده بهینه از منابع میزبان می‌شود و اگر بسیار دیر اعلام شود هم باعث وقوع سرریز، خطا در روند محاسبات و نقض

SLA می‌شود. یکی از مراحل که مکانیسم تحمل‌پذیری خطا به کار برده می‌شود این مرحله است که در ادامه تشریح شده است.

در تحقیقات گذشته روش‌های متفاوتی برای این منظور به کار رفته است. در این مرحله ما برای شناسایی میزبان‌های زیر بار کاری بیشتر از روش میانگین انحراف معیار بهره برده‌ایم. این روش برای تعیین حد آستانه سازگار (پویا) و جلوگیری از نقض سطح توافق سرویس است. این یک روش آماری برای سنجش پراکندگی داده‌ها و یک شیوه هموارسازی داده‌ها می‌باشد. این روش با متوسط‌گیری از یافته‌های متوالی در هر لحظه زمان، داده‌ها را از طریق کاهش اختلال و حذف عوامل اضافی (noise) در مشاهدات هموار می‌کند. این الگوریتم نسبت به انحراف استاندارد مقاوم بوده و از یافته‌های قبلی برای تولید ارزش آستانه جدید استفاده می‌کند.

همان‌طور که می‌دانیم اختلاف هرکدام از اعداد نسبت به میانگین، انحراف است. البته برای انحراف می‌توانیم مقدار قدر مطلق را در نظر بگیریم و آنها را مثبت فرض کنیم. اگر میانگین همه این انحراف‌ها را به دست بیاوریم؛ معیاری استاندارد به دست می‌آید که این معیار جدید همان میانگین قدر مطلق انحراف خواهد بود. این روش راهکاری برای به‌دست‌آوردن ایده کلی از روند مجموعه داده‌ها است. این میانگین برای پیش‌بینی روندهای بلندمدت بسیار مفید بوده و می‌توان برای هر دوره زمانی محاسبه نمود. از رابطه ۱۸ برای محاسبه این معیار استفاده می‌شود.

$$MAD = \text{median}_i(X_i - \text{median}_i(X_j)) \quad (19)$$

در این روش، حد آستانه توسط رابطه ۱۹ تعیین می‌شود که در آن هرچه مقدار S کمتر باشد، مصرف انرژی نیز کمتر شده و نقض SLA افزایش می‌یابد؛ لذا باید تلاش شود تا توازن لازم همیشه برقرار باشد.

$$T_U = 1 - S.MAD \quad (20)$$

۳-۳-۴- مدیریت بار کاری برای رسیدن به تعادل

در این مرحله این مسئله بررسی می‌شود که به چه میزان از حجم بار کاری میزبان فیزیکی کم کنیم تا تعادل برقرار شود. انتخاب یک ماشین مجازی که از منابع میزبان زیاد استفاده می‌کند باعث کاهش کارایی میزبان مربوطه و افزایش هزینه مهاجرت می‌شود. اگر ماشین مجازی که استفاده کمینه از منابع میزبان فیزیکی را دارد را انتخاب کنیم باعث می‌شود میزبان در حالت ریسک‌پذیر سرریز بماند و مجبور به انتخاب‌های متوالی شده و مهاجرت‌های زیادی را در پی

داشته باشیم که هم زمان‌بر و هم هزینه‌بر خواهد بود. بنابراین رعایت این توازن مهم و حیاتی خواهد بود. این کار نیز با استفاده از الگوریتم جستجوی تصادفی^۱ انجام می‌گیرد. برخی الگوریتم‌ها از اعداد تصادفی برای تصمیم‌گیری در مورد کار بعدی در هر مکانی در منطق خود استفاده می‌کنند. برای نمونه، در مرتب‌سازی سریع تصادفی، از شماره تصادفی برای انتخاب محور بعدی استفاده می‌کنیم یا آرایه را به طور تصادفی تغییر می‌دهیم. در اینجا مرحله انتخاب ماشین مجازی زمانی اجرا می‌شود که یک میزبان پربار تشخیص داده شده است. در این روش به صورت کاملاً تصادفی یک ماشین مجازی از روی میزبانی که دچار سرریز شده، انتخاب و مهاجرت داده می‌شود. اگر میزبان با این انتخاب از حالت سرریز خارج نشود دوباره الگوریتم فراخوانی شده و یک ماشین مجازی دیگر برای مهاجرت انتخاب می‌شود.

۳-۳-۵- مهاجرت ماشین‌های مجازی منابع زیربار

در این مرحله، برای تصمیم‌گیری در مورد مهاجرت ماشین‌های مجازی از میزبان‌های پربار به میزبان‌های کم‌بار، از پارامتر IFS^2 در قوانین ریاضی فرکتال برای پیش‌بینی دقیق بار میزبان‌های مقصد استفاده می‌شود. این رویکرد فرکتال، با تحلیل الگوهای پیچیده و خودمتشابه در سری‌های زمانی بار کاری، امکان پیش‌بینی وضعیت آتی میزبان‌ها را با دقت بالا فراهم می‌کند. به طور خاص، اگر پیش‌بینی بار میزبان بعدی بر اساس تحلیل فرکتال، از آستانه ۸۵ درصد فراتر رود، آن میزبان فیزیکی به عنوان پربار شناسایی شده و از انتخاب به عنوان مقصد مهاجرت حذف می‌شود. این استفاده از IFS به ما اجازه می‌دهد تا از وقوع سرریز جلوگیری کرده و تصمیمات مهاجرت را بر اساس یک مدل پیش‌بینی هوشمند و پویا اتخاذ کنیم، که به حفظ تعادل بار و بهینه‌سازی مصرف انرژی کمک شایانی می‌کند.

رویکردی که برای مهاجرت ماشین‌های مجازی هر میزبان در نظر گرفته شده است از ۴ بخش اصلی تشکیل شده است: (۱) گزینش گره مبدأ (۲) گزینش ماشین‌های مجازی (۳) گزینش گره مقصد (۴) مقادیر به‌روز شده فاصله. این مقاله در این مرحله از روش پیشنهادی در نظر دارد بهره‌وری انرژی را از طریق بهینه‌سازی مهاجرت ماشین‌های مجازی به حداکثر برساند و بدان وسیله سطوح استفاده از منابع را بهبود بخشد. رویکرد پیشنهادی برای مهاجرت ماشین‌های مجازی به گره‌ها برای صرفه‌جویی در انرژی با استفاده از منابع و حفظ کارایی با به حداقل رساندن کارها، مورد استفاده قرار می‌گیرد. مسئله با بهینه‌سازی مصرف انرژی (EC) و همچنین زمان اجرا (ET) از طریق تابع برازندگی بهینه می‌گردد. مقادیر کوچک EC نشان می‌دهد که گره مربوط به آن دارای مصرف انرژی کمتری می‌باشد. ET یکی دیگر

^۱ Random search

^۲ Iterated Function System

از معیارهای بهینه‌سازی است که اشاره به کارهای در حال اجرا در ماشین مجازی در هر گره دارد. مسئله برای رسیدن به یک راه‌حل بهینه به صورت زیر فرمول شده است.

برای در نظر گرفتن این مسئله، مجموعه‌ای از گره‌ها $\{r_1, r_2, \dots, r_m\}$ در نظر گرفته شده است، به طوری که $R = \{r_i | 1 \leq i \leq M\}$ مجموعه‌ای از M گره می‌باشد، یک مجموعه از $VM = \{v_j | 1 \leq j \leq N\}$ به طوری که $\{v_1, v_2, \dots, v_n\}$ ماشین مجازی می‌باشد؛ یک مجموعه از کارها $\{job_1, job_2, \dots, job_l\}$ برای اجرا در ماشین‌های مجازی به طوری که $JOB = \{job_k | 1 \leq k \leq L\}$ می‌باشد. c_i ظرفیت کلی پردازنده در میلیون دستورالعمل در هر ثانیه (MIPS) و m_i ظرفیت حافظه در MB از گره r_i می‌باشد. vc_j و vm_j ظرفیت‌های حافظه و پردازنده v_j می‌باشند.

• گام اول - گزینش گره مبدأ

گره مبدأ گره فعالی می‌باشد که از آن ماشین‌های مجازی باید مهاجرت کنند. گره فعالی که در حداقل فاصله از گره مقصد است به عنوان گره مبدأ انتخاب می‌شود. در روش پیشنهادی برای گزینش این گره از الگوریتم کاپوچین استفاده شده است. مرحله‌های الگوریتم کاپوچین برای این گام در زیر تشریح شده است.

○ الف- مرحله اول الگوریتم: آماده‌سازی الگوریتم و انجام محاسبات اولیه برای تشکیل راه‌حل‌های اولیه

در این مرحله در ابتدا مقدار مصرف انرژی برای هر گره فعال با استفاده از رابطه ۴ محاسبه و سپس در یک لیست همه مقادیر ذخیره گردیده است. همچنین بهره‌وری مبتنی بر زمان برای هر کدام از گره‌ها محاسبه می‌شود به این صورت که پس از محاسبه EC هر گره فعال، در این مرحله بهینه‌سازی عملکرد از نظر به حداقل رساندن زمان اجرای کارها بررسی می‌شود. ET برای هر گره فعال بر اساس رابطه ۶ محاسبه شده و مقادیر مجدداً در لیست ذخیره می‌شوند.

○ ب- مرحله دوم الگوریتم کاپوچین [40] تولید راه‌حل برای یافتن منبع غذا

در این مرحله کاپوچین‌ها برای تولید جواب از الگوی حرکتی خود استفاده می‌کنند. از این رو، با مشخص نمودن مناطق سرشار از مواد غذایی شاخصی به عنوان FFI^1 تعریف می‌شود. این شاخص برای هر منبع که به عنوان گره فعال می‌باشد بر اساس انرژی و زمان انجام حجم کارها محاسبه می‌شود. جستجوی کاپوچین‌ها برای پیدا نمودن نواحی سرشار از مواد غذایی یک رویه بهینه‌سازی می‌باشد؛ بنابراین مقدار FFI با استفاده از شاخص‌های EC و ET محاسبه شده و یک لیست مرتب از FFI بر اساس مقادیر ارائه می‌گردد. گره فعال با حداقل EC و ET به عنوان عنصر اول لیست به دست آمده است.

¹ Find Food Index

○ پ- مرحله سوم الگوریتم کاپوچین - ارزیابی راه‌حل‌ها

مرحله ارزیابی راه‌حل‌های کاپوچین برای مسئله مهاجرت ماشین‌های مجازی بر اساس پارامتری به‌عنوان فاصله صورت می‌گیرد. به این صورت که در این مرحله گرهی به‌عنوان یک منبع بالقوه برای مهاجرت ماشین‌های انتخاب می‌شود که معیار کمینه مصرف انرژی را برآورده سازد. این معیار توسط پارامتر فاصله و با استفاده از رابطه ۲۰ محاسبه می‌شود.

$$\text{Distance} = \text{Avg} (\text{FFI}_{\text{mid}}, \text{FFI}_{\text{max}}) \quad (21)$$

گره‌ی که دارای مقدار EC نزدیک به مقدار فاصله محاسبه‌شده باشد، برای گره مبدأ بودن؛ جایی که ماشین‌های مجازی از آنجا منتقل خواهند شد، مناسب می‌باشد. در این مرحله گرهی که برای مهاجرت انتخاب‌شده است، در تکرار بعدی برای مهاجرت حذف خواهد شد.

• گام دوم - گزینش ماشین‌های مجازی

در این مرحله، ماشین‌های مجازی برای مهاجرت کردن تعیین می‌شوند. برای انتخاب ماشین‌های مجازی برای منتقل شدن از گره مبدأ، بار هر ماشین مجازی در گره مبدأ با توجه به رابطه ۲۱ محاسبه می‌شود. سپس این مقادیر در یک لیست ذخیره می‌شود و لیست از مقادیر بار بالا به پایین به‌صورت نزولی مرتب می‌شود و ماشین مجازی با مقدار بار بالا برای انتقال به گره مقصد انتخاب می‌شود.

$$\text{load}_{ij} = \frac{\sum_{j=1}^v \text{job}_j}{\left(\frac{\sum_{j=1}^v \text{rpu}_{ij} \sum_{j=1}^v \text{rmu}_{ij}}{c} \right) \Delta t} \quad (22)$$

در رابطه ۲۱، job_j نشان‌دهنده تعداد کل وظایف در ماشین مجازی j ام در گره i ام و $\left(\frac{\sum_{j=1}^v \text{cpu}_{ij} \sum_{j=1}^v \text{mu}_{ij}}{c} \right)$ کل برق مصرفی

گره i ام در واحد زمان Δt می‌باشد.

• گام سوم - گزینش گره مقصد

در این مرحله، گره مقصد برای ماشین‌های مجازی پیدا می‌شوند. پیدا کردن ناحیه پر از مواد غذایی که نیازمند پیدا کردن گره پر از مواد غذایی می‌باشد. اگر مصرف انرژی گرهی کمینه باشد آن گره سرشار از مواد غذایی تلقی

می‌شود. گره فعال با حداقل EC به عنوان عنصر اول لیست FFI به دست آمده است و در نتیجه آن گره مدعی اصلی برای مهاجرت ماشین‌های مجازی بیش از حد بارگذاری شده به آن است.

• گام چهارم - به روزرسانی فاصله

به روزرسانی درگیر شده پس از هر تکرار، مقدار به روز شده برای فاصله می‌باشد. هرگاه که ماشین مجازی در حال اجرا نیازمند مهاجرت باشد کل وضعیت ماشین مجازی شامل پردازنده‌های مجازی، پیکربندی درایوها، حافظه ماشین مجازی، ذخیره سازی نقل مکان می‌شوند. برای مهاجرت ماشین‌های مجازی به صورت کارآمد و قابل دسترسی بودن، مقادیر و داده‌های همه ماشین‌های مجازی بر روی یک محل ذخیره سازی مشترک^۱، ذخیره می‌شود. سرور ذخیره سازی مشترک برای کلیه گره‌ها و سرورها به عنوان محل ذخیره سازی برای ماشین‌های مجازی که در آن داده‌های غیر زائد ذخیره شده در دسترس است. هرگاه یک ماشین مجازی منتقل شود، تنها حافظه داخلی و رجیسترهای پردازنده آن ماشین مجازی نیاز به مهاجرت از آن گره به گره دیگر دارند، به طوری که تصاویر و محتویات ذخیره سازی از سرور ذخیره سازی مشترک قابل دسترسی است. این کار موجب صرفه جویی فضای ذخیره سازی می‌شود، بهبود نرخ جست و جو و همچنین بهبود نرخ جریان داده در زمان مهاجرت ماشین مجازی می‌شود. علاوه بر این، به کاهش نقل و انتقالات داده و پس از آن به کاهش زمان مصرف شده برای مهاجرت ماشین مجازی کمک می‌کند. اجرای سریع تر وظایف منجر به حداقل رساندن زمان اجرای کل سیستم می‌شود و در ادامه باعث بهبود عملکرد می‌شود.

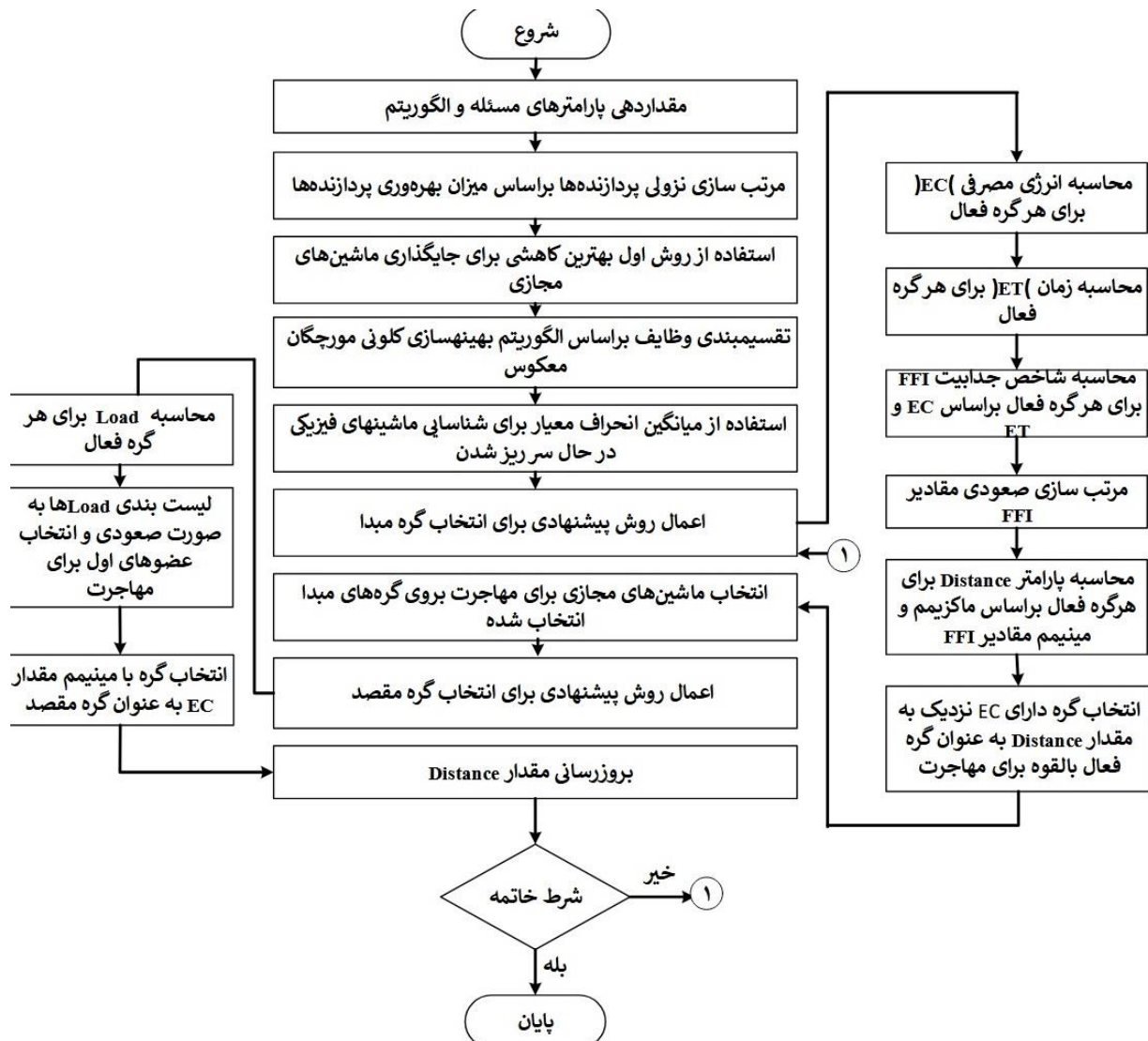
۳-۳-۶- بررسی کردن میزان سرریزی مجدد

در این مرحله نیز برای تعیین دقیق تر حد آستانه سازگار و پویا در میزبان‌هایی که پربار بوده و در آستانه سرریزی هستند از روش میانگین انحراف معیار استفاده می‌شود که در مرحله ۳ توضیح داده شد. چنانچه بار یک میزبان از یک حد آستانه فراتر رود؛ ماشین مجازی به میزبان بعدی فرستاده می‌شود. چنانچه ماشین فیزیکی بعدی نیز گنجایش لازم را نداشته و پربار بود ماشین مجازی دوباره به فهرست Vm Migrate-able بازگشته و در صف قرار می‌گیرد که سپس بر اساس اولویت و به صورت FIFO روند تخصیص دوباره انجام می‌شود.

^۱ Network attached storage (NAS)

۳-۳-۷- بررسی داده‌های ارسالی در جهت بهینه‌سازی انرژی

برای دستیابی به بهینه‌سازی انرژی و تجمیع مؤثر ماشین‌های مجازی، از قوانین ریاضیاتی فرکتال برای شناسایی و کاهش ارسال داده‌های تکراری استفاده می‌شود. این رویکرد، با بهره‌گیری از خاصیت خودمتمابه‌ی و تکرارپذیری الگوها در جریان داده‌ها، امکان فیلتر کردن و حذف داده‌های زائد را فراهم می‌آورد. با کاهش حجم داده‌های تکراری و هدایت داده‌های باقی‌مانده به نزدیک‌ترین استخر منابع، ماشین‌های مجازی با حجم داده‌های تکراری کمتر، به طور هوشمندانه در یک میزبان فیزیکی تجمیع می‌شوند. این تجمیع، منجر به افزایش بهره‌وری و پر شدن ظرفیت میزبان‌های فیزیکی می‌شود. در نتیجه، با پر شدن ظرفیت میزبان‌های فعال و خالی ماندن ظرفیت میزبان‌های بعدی، امکان خاموش کردن میزبان‌های بلااستفاده فراهم می‌آید. این مکانیزم مبتنی بر فرکتال، به طور مستقیم به کاهش مصرف انرژی کلی سیستم کمک می‌کند، زیرا میزبان‌های خاموش شده دیگر انرژی مصرف نمی‌کنند و مهاجرت‌های جدید نیز به حداقل می‌رسند. با کاهش ارسال داده‌های تکراری به روش قوانین ریاضیاتی فرکتال و ارسال داده‌ها به نزدیک‌ترین استخر، ماشین‌های مجازی زیادی که دارای داده‌های تکراری کمتری هستند در یک میزبان تجمیع می‌شوند تا آن ماشین فیزیکی نیز به ظرفیت لازم دست‌یافته و پربار شود. با توجه به ادامه روند پر شدن گنجایش ماشین فیزیکی و خالی ماندن ظرفیت میزبان بعدی؛ ظرفیت میزبان آخر کاملاً خالی شده و نیازی به روشن ماندن آن نیست. با خاموش شدن آخرین میزبان و تجمیع ماشین‌های مجازی در میزبان‌های قبلی، مهاجرت جدیدی نیز انجام نمی‌شود. بنابراین میزبان خاموش شده و مصرف انرژی نیز با توجه به زیر بار نبودن میزبان، بدین ترتیب کاهش خواهد یافت. فلوچارت روش پیشنهادی در شکل ۲ آورده شده است.



شکل ۲: فلوچارت روش پیشنهادی

۴- ارزیابی نتایج بدست آمده

در این بخش نتایج شبیه‌سازی روش پیشنهادی در چند سناریو و آزمایش آورده شده است. برای پیاده‌سازی روش پیشنهادی از شبیه‌ساز متلب با مشخصات سخت‌افزاری که در جدول ۱ مشخص شده استفاده شده است. به‌طورکلی برای ارزیابی روش پیشنهادی دو سناریو در نظر گرفته شده است. در سناریو اول عملکرد روش پیشنهادی برای حالت‌های بدون و با مهاجرت ماشین‌های مجازی و اعمال روش پیشنهادی در این مهاجرت بررسی شده است و در ادامه در سناریو دوم، عملکرد روش پیشنهادی با کارهای پیشین مقایسه شده است.

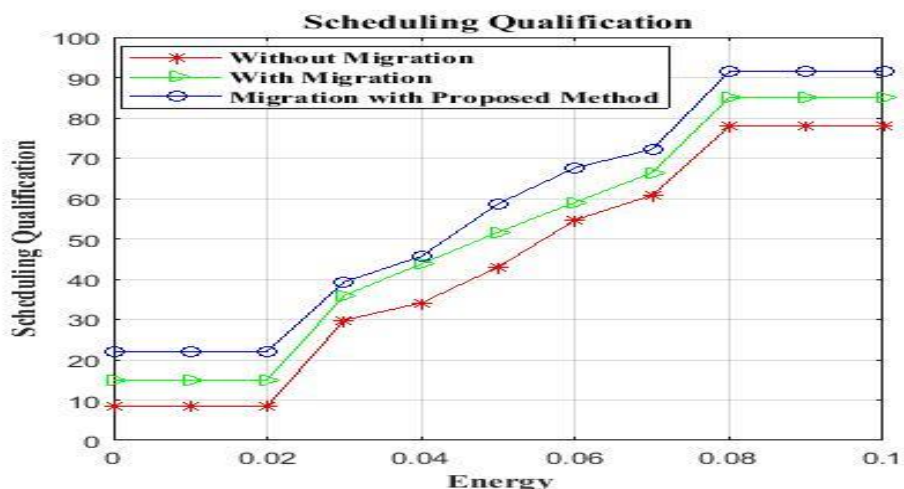
جدول ۱: مشخصات سخت‌افزاری

مؤلفه	مشخصات
سیستم عامل	ویندوز هشت ۶۴ بیتی
پردازشگر	Intel core i7
حافظه رم	۸ گیگ
حافظه کارت گرافیک	۱ گیگ

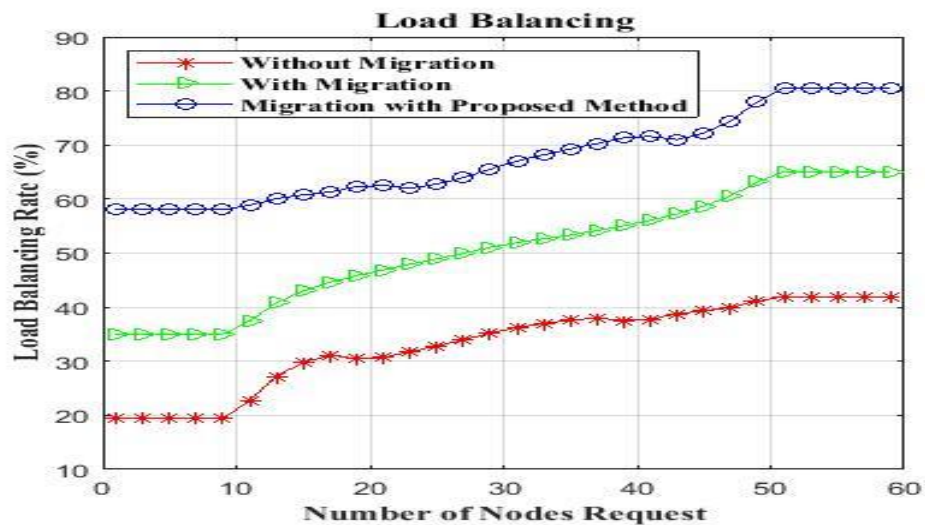
۴-۱- سناریو اول: ارزیابی عملکرد روش پیشنهادی با در نظر گرفتن مهاجرت و بدون مهاجرت

از آنجائی که در روش پیشنهادی از مهاجرت ماشین‌های مجازی برای بهبود انرژی مصرفی ماشین‌های فیزیکی و بهبود تعادل بار بین ماشین‌ها با استفاده از زمان‌بندی بهینه بین منابع استفاده شده است، از این‌رو در این سناریو عملکرد روش ترکیبی را با در نظر گرفتن مهاجرت و بدون در نظر گرفتن آن مورد ارزیابی قرار داده‌ایم. همان‌طور که در شکل‌های ۳ تا ۸ مشخص است، استفاده از یک الگوی مهاجرت بهینه، در کنار کاهش مصرف انرژی منابع، کیفیت زمان‌بندی و تعادل بار را افزایش داده است. علاوه بر این، میزان تحمل‌پذیری خطا با افزایش میزان درخواست‌های کاربر، افزایش یافته و منابع می‌توانند با کم‌ترین نرخ خرابی و خطا، در زمان‌بندی وظایف نقش داشته باشند. این سناریو در دو آزمایش مختلف با تعداد متفاوت درخواست‌ها و ماشین‌های مجازی مقایسه انجام شده است.

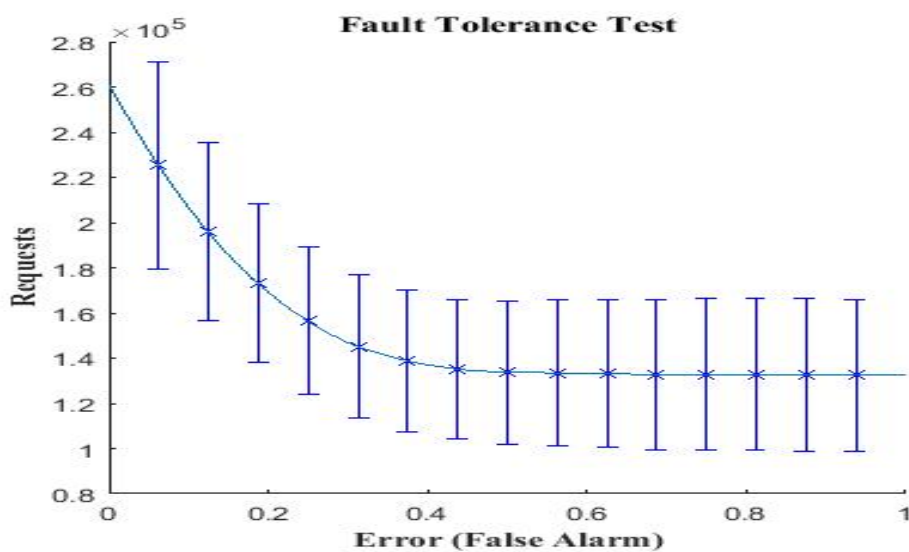
آزمایش اول: بررسی کیفیت زمان‌بندی، تعادل بار و نرخ خطا با تعداد ۲۰ ماشین فیزیکی و ۳۵ ماشین مجازی در نمودارهای شکل ۳ و ۴ منحنی سبز رنگ مهاجرت برپایه الگوریتم مورچگان معکوس، منحنی آبی رنگ بدون استفاده از الگوریتم‌های پیشنهادی و مورچگان معکوس را نشان می‌دهد.



شکل ۳: بررسی کیفیت زمان‌بندی با در نظر گرفتن ۲۰ ماشین فیزیکی و ۳۵ ماشین مجازی

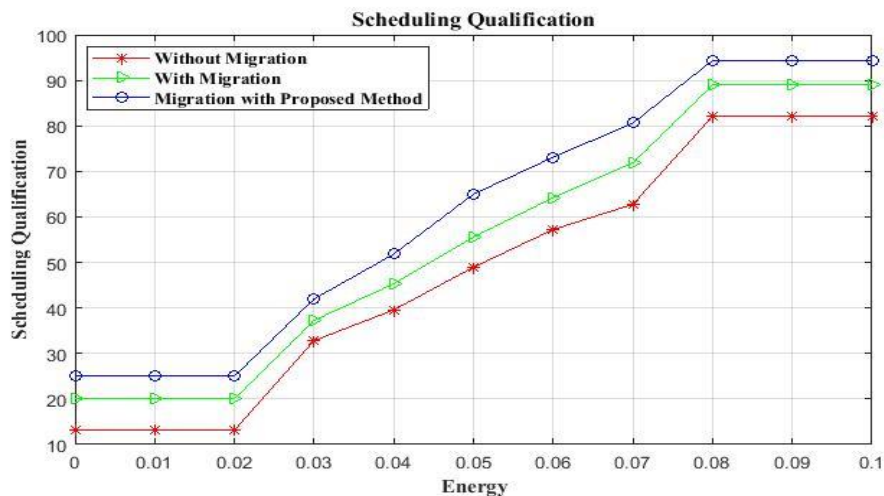


شکل ۴: بررسی تعادل بار با در نظر گرفتن ۲۰ ماشین فیزیکی و ۳۵ ماشین مجازی



شکل ۵: بررسی نرخ خطا با در نظر گرفتن ۲۰ ماشین فیزیکی و ۳۵ ماشین مجازی

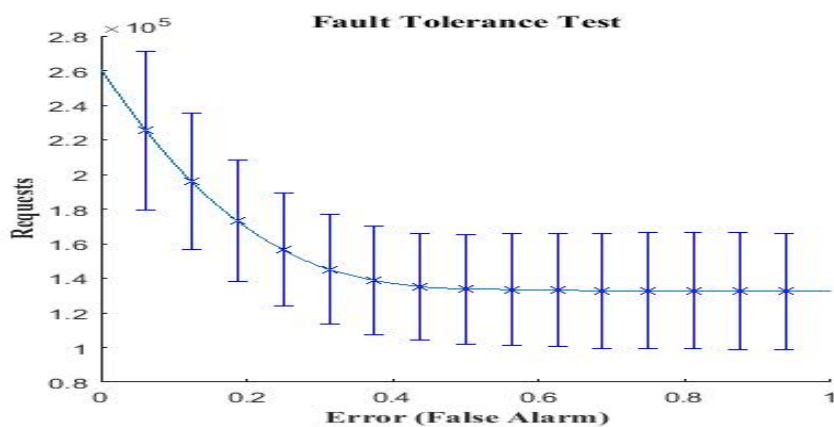
آزمایش دوم: بررسی کیفیت زمان‌بندی، تعادل بار و نرخ خطا با تعداد ۳۰ ماشین فیزیکی و ۴۵ ماشین مجازی



شکل ۶: بررسی کیفیت زمان‌بندی با در نظر گرفتن ۳۰ ماشین فیزیکی و ۴۵ ماشین مجازی



شکل ۷: بررسی تعادل بار با در نظر گرفتن ۳۰ ماشین فیزیکی و ۴۵ ماشین مجازی



شکل ۸: بررسی نرخ خطا با در نظر گرفتن ۳۰ ماشین فیزیکی و ۴۵ ماشین مجازی

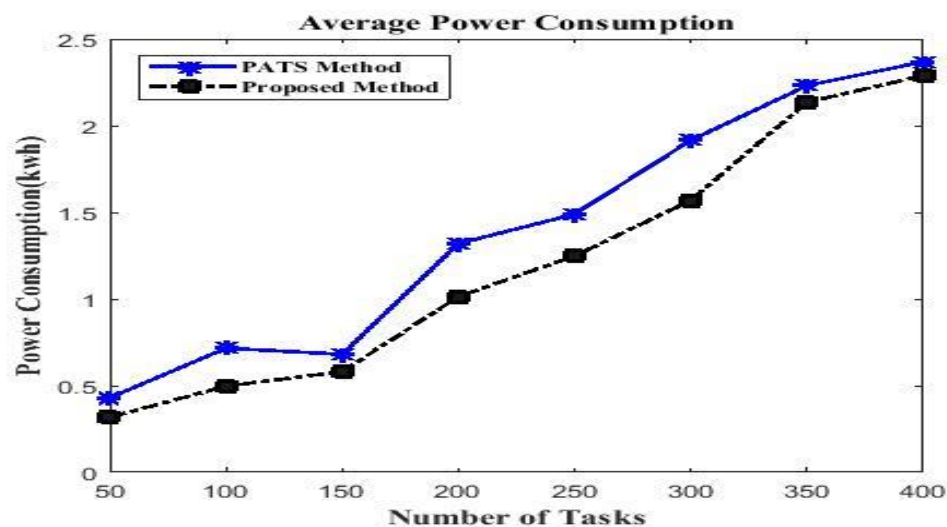
۴-۲- سناریو دوم: ارزیابی عملکرد روش پیشنهادی در مقایسه با سایر روش‌های ارائه شده

در هرکدام از آزمایش‌های تدوین شده، عملکرد روش پیشنهادی با روش‌های پیشین مقایسه شده است.

آزمایش اول: ارزیابی عملکرد روش پیشنهادی و مقایسه آن با روش ارائه شده در [41]

در این آزمایش میزان مصرف انرژی روش پیشنهادی متناسب با داده‌های مقاله [41] ارزیابی شده و با روش مقاله،

یعنی PATS مقایسه شده است. در این آزمایش از تعداد ۵ ماشین فیزیکی و ۲۰ ماشین مجازی و تعداد ۵۰ تا ۴۰۰ وظیفه استفاده شده است و نتیجه بدست آمده در شکل ۹ آورده شده است.



شکل ۹: میزان انرژی مصرفی در روش پیشنهادی و مقایسه آن با روش PATS

آزمایش دوم: ارزیابی عملکرد روش پیشنهادی و مقایسه آن با روش ارائه شده در [42]

در این آزمایش عملکرد روش پیشنهادی از نظر پارامترهای انرژی و زمان اجرا با روش ارائه شده در مقاله [42]

یعنی روش FCHS مقایسه شده است. همان‌طور که از نتایج بدست آمده مشخص است، روش پیشنهادی توانسته است

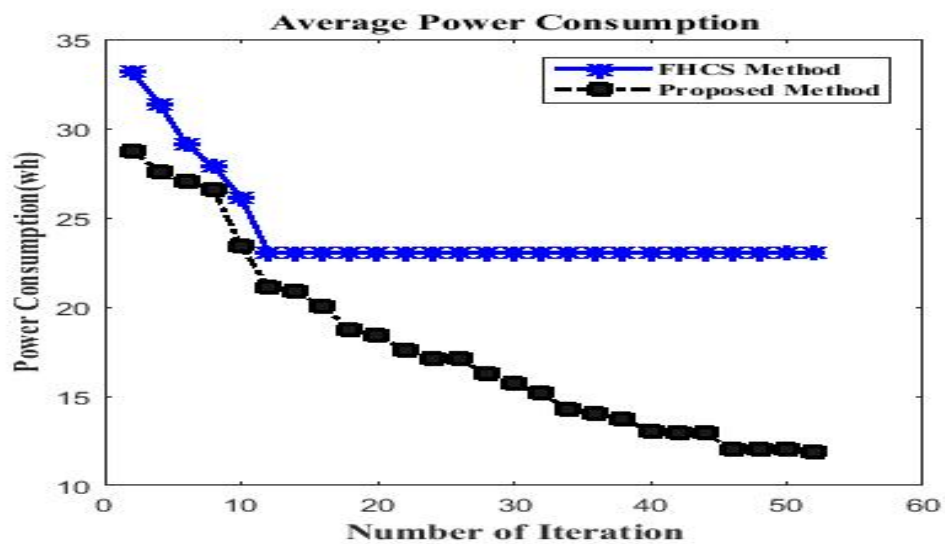
از نظر انرژی به نسبت به روش FCHS میزان انرژی کمتری مصرف کند که دلیل این امر را می‌توان عملکرد مناسب

روش پیشنهادی در مهاجرت دادن وظایف ماشین‌های مجازی زیر بار و انجام همزمان زمان‌بندی بیان کرد. در این

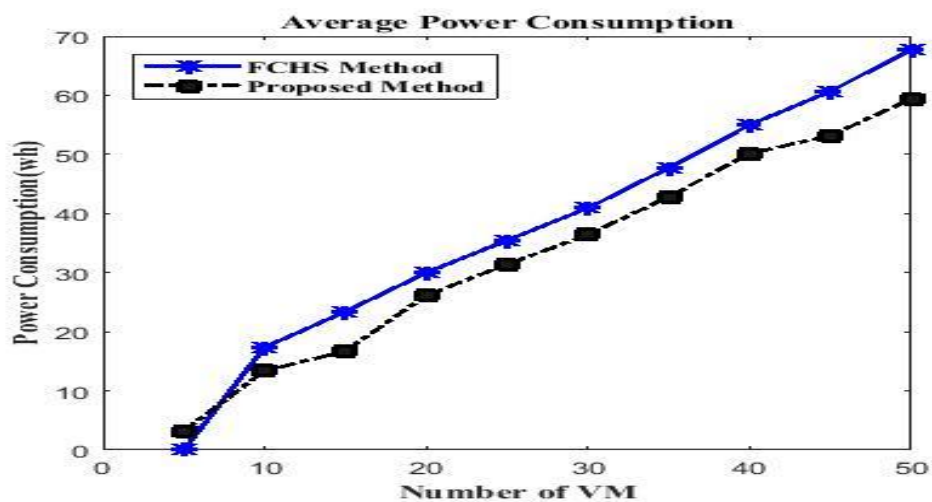
آزمایش از تعداد ۵ تا ۵۰ ماشین مجازی و ۴ ماشین فیزیکی استفاده شده است. همچنین نتایج بدست آمده برای میزان

انرژی مصرفی براساس تعداد تکرار و همچنین تعداد ماشین‌های مجازی به ترتیب در شکل‌های ۱۰ و ۱۱ و ارزیابی

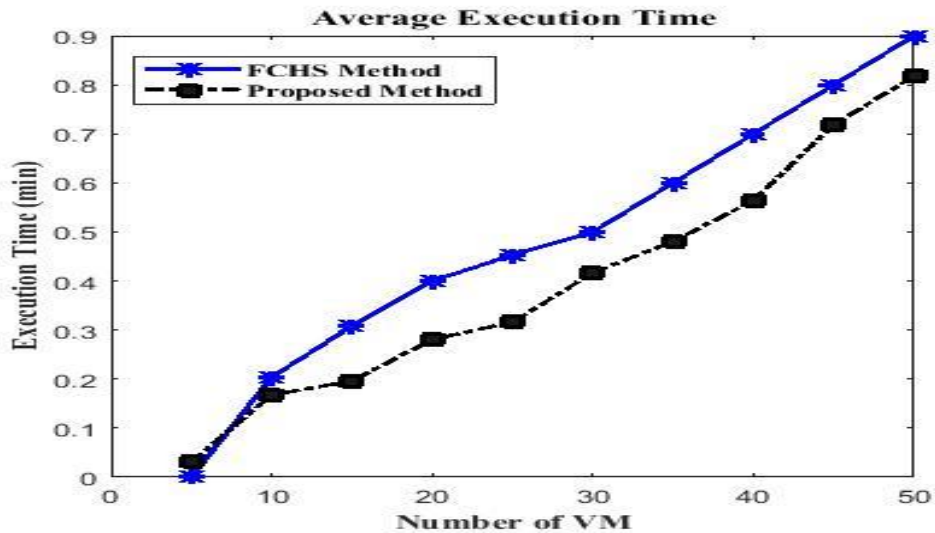
زمان اجرای روش پیشنهادی در شکل ۱۲ آورده شده است.



شکل ۱۰: میزان انرژی مصرفی در روش پیشنهادی براساس تعداد تکرار الگوریتم و مقایسه آن با روش FHCS



شکل ۱۱: میزان انرژی مصرفی در روش پیشنهادی براساس تعداد ماشین‌های مجازی و مقایسه آن با روش FHCS



شکل ۱۲) میزان زمان اجرا در روش پیشنهادی و مقایسه آن با روش FCHS

آزمایش سوم: ارزیابی عملکرد روش پیشنهادی و مقایسه آن با روش ارائه شده در [43]

در این آزمایش، عملکرد روش پیشنهادی از نظر پارامترهای Makespan، انرژی و زمان مهاجرت با روش ارائه شده در مقاله [43] که یک روش چندهدفه براساس الگوریتم هوش ازدحامی گربه است، مقایسه شده است. در این آزمایش برای ارزیابی روش پیشنهادی از workloadهای HPC2N و NASA استفاده شده است. در این آزمایش از ۵۰۰ ماشین فیزیکی و ۲۰۰ ماشین مجازی و تعداد ۱۰۰ تا ۱۰۰۰ وظیفه استفاده شده است. همانطور که در شکل‌های ۱۳-۱۷ نشان داده شده است روش پیشنهادی در این مقاله از نظر پارامترهای اشاره شده بهتر از روش CSO عمل می‌کند.

برای ارزیابی زمان تکمیل با استفاده از بار کاری HPC2N، ما تحلیلی را انجام داده‌ایم که شامل مجموع ۱۰۰۰ وظیفه است. همانطور که در شکل ۱۵ و جدول ۲ نشان داده شده است، برای سناریویی شامل ۲۰۰ وظیفه، زمان تکمیل محاسبه شده برای رویکرد CSO برابر با ۱۲۵۱٫۳ است، در حالی که الگوریتم پیشنهادی زمان تکمیل ۱۲۱۱٫۱ را به دست می‌دهد. در مورد ۶۰۰ وظیفه، زمان تکمیل به دست آمده با استفاده از تکنیک CSO برابر با ۱۵۴۷٫۲ اندازه‌گیری شده است، در حالی که الگوریتم پیشنهادی به زمان تکمیل ۱۵۰۵٫۳ دست می‌یابد. با گسترش ارزیابی به ۱۰۰۰ وظیفه، روش CSO زمان تکمیل ۱۹۸۵٫۹ را تولید می‌کند، در حالی که الگوریتم ما زمان تکمیل ۱۹۰۴٫۵ را ارائه می‌دهد. برای ارزیابی زمان تکمیل با استفاده از بار کاری ناسا، ارزیابی ما شامل ۱۰۰۰ وظیفه اجرا شده برای اهداف شبیه‌سازی است. نتایج، همانطور که در شکل ۱۶ و جدول ۳ نشان داده شده است، به شرح زیر است. برای یک حجم کاری شامل ۲۰۰

وظیفه، روش CSO زمان انجام ۶۴۱,۴ را ارائه می‌دهد، در حالی که الگوریتم پیشنهادی زمان انجام کاهش یافته ۶۲۰,۲ را نشان می‌دهد. در زمینه ۶۰۰ وظیفه، زمان انجام ثبت شده برای رویکرد CSO به ۸۷۸,۵ می‌رسد، در حالی که الگوریتم ما زمان انجام بهبود یافته قابل توجهی به ۸۱۹,۴ را نشان می‌دهد. با گسترش ارزیابی به ۱۰۰۰ وظیفه، الگوریتم CSO زمان انجام ۹۴۵,۶ را ارائه می‌دهد، در حالی که الگوریتم ما زمان انجام کاهش یافته ۸۸۱,۳ را تولید می‌کند. از این نتایج می‌توان تشخیص داد که الگوریتم ما در مقایسه با الگوریتم CSO، بهبود قابل توجهی در عملکرد زمان انجام ارائه می‌دهد.

EC به عنوان یک پارامتر محوری با پیامدهای گسترده برای ارائه‌دهندگان ابر و کاربران مطرح است. ضرورت به حداقل رساندن مصرف انرژی، یک چالش اصلی در الگوی محاسبات ابری است. با دستیابی به بهره‌وری انرژی، ارائه‌دهندگان ابر می‌توانند هزینه‌های انرژی خود را کاهش دهند و در عین حال وظایف روی منابع مجازی را به طور مؤثر مدیریت کنند. همزمان، کاربران ابر می‌توانند از در دسترس بودن منابع با هزینه‌های کمتر بهره‌مند شوند. در زمینه ارزیابی مصرف انرژی با استفاده از بار کاری HPC2N، ارزیابی ما شامل ۱۰۰۰ وظیفه اجرا شده برای اهداف شبیه‌سازی است. نتایج، همان‌طور که در جدول ۴ و شکل ۱۷ ارائه شده است، نشان می‌دهد که برای یک بار کاری شامل ۲۰۰ وظیفه، رویکرد CSO 45.9 واحد انرژی مصرف می‌کند، در حالی که الگوریتم ما کاهش مصرف انرژی ۴۳,۹ واحد را نشان می‌دهد. با در نظر گرفتن ۶۰۰ وظیفه، انرژی مصرف شده توسط روش CSO به ۶۷,۷ واحد می‌رسد، در حالی که الگوریتم ما به کاهش مصرف انرژی ۶۳,۵ واحد دست می‌یابد. با گسترش ارزیابی به ۱۰۰۰ وظیفه، الگوریتم CSO منجر به مصرف انرژی ۱۱۳,۹ واحد می‌شود، در حالی که مصرف انرژی الگوریتم ما ۹۰,۳ واحد کمتر است. در زمینه ارزیابی مصرف انرژی با استفاده از آرشیو حجم کار ناسا، نتایج ارزیابی، همان‌طور که در جدول ۵ و شکل ۱۸ نشان داده شده است، نشان می‌دهد که برای یک حجم کار شامل ۲۰۰ وظیفه، رویکرد CSO مصرف انرژی ۳۲,۳ واحد را ثبت می‌کند، در حالی که الگوریتم ما مصرف انرژی کاهش یافته ۳۰,۷ واحد را نشان می‌دهد. با در نظر گرفتن ۶۰۰ وظیفه، مصرف انرژی مرتبط با روش CSO برابر با ۴۵,۶ واحد است، در حالی که الگوریتم ما مصرف انرژی کاهش یافته ۴۳,۱ واحد را نشان می‌دهد. با گسترش ارزیابی به ۱۰۰۰ وظیفه، الگوریتم CSO مصرف انرژی ۱۰۰,۴ واحد را ثبت می‌کند، در حالی که مصرف انرژی الگوریتم ما ۷۹,۱ واحد کمتر است. یافته‌ها، کاهش قابل توجه مصرف انرژی انجام شده توسط الگوریتم پیشنهادی ما را در مقایسه با رویکرد CSO نشان می‌دهند.

ارزیابی زمان مهاجرت با استفاده از حجم کار HPC2N شامل مجموع ۱۰۰۰ وظیفه برای شبیه‌سازی بود. نتایج ارائه شده در جدول ۶ و شکل ۱۹ نشان می‌دهد که برای یک حجم کاری شامل ۲۰۰ وظیفه، زمان مهاجرت برای رویکرد CSO برابر با ۱۰۱۴,۲ واحد و برای الگوریتم ما برابر با ۱۰۰۱,۳ واحد اندازه‌گیری شده است. با در نظر گرفتن ۶۰۰

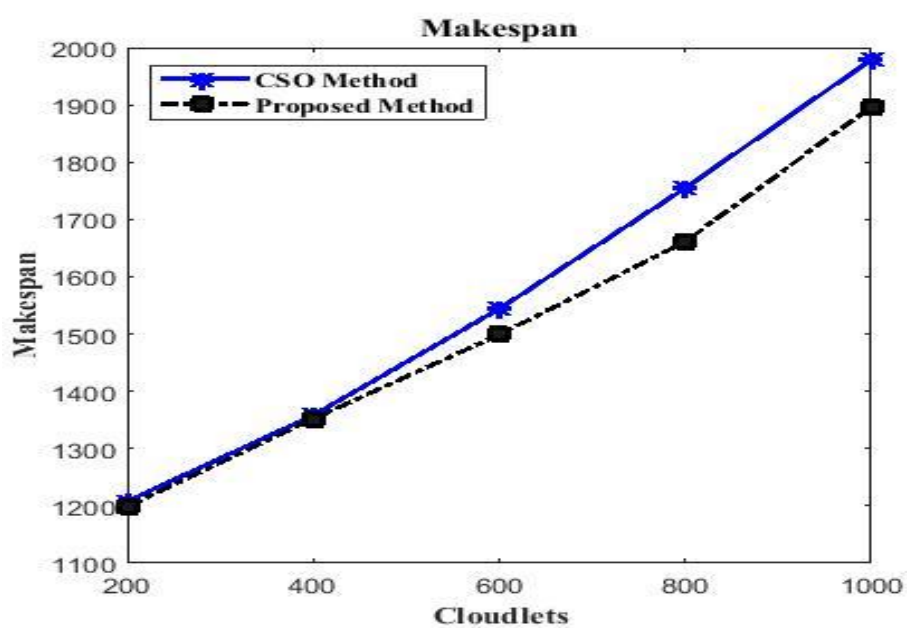
وظیفه، زمان مهاجرت برای روش CSO برابر با ۱۱۴۸.۴ واحد و برای الگوریتم ما برابر با ۱۱۲۹.۳ واحد است. با گسترش ارزیابی به ۱۰۰۰ وظیفه، زمان مهاجرت برای الگوریتم CSO برابر با ۱۴۳۲.۸ واحد و برای الگوریتم ما برابر با ۱۴۱۷.۹ واحد خواهد بود. علاوه بر این، همان‌طور که در جدول ۷ و شکل ۲۰ نشان داده شده است، ارزیابی زمان مهاجرت با استفاده از حجم کاری ناسا نشان می‌دهد که برای یک حجم کاری شامل ۲۰۰ وظیفه، زمان مهاجرت برای رویکرد CSO برابر با ۵۵۵.۴ واحد و برای الگوریتم ما برابر با ۵۴۸.۷ واحد ثبت شده است. با در نظر گرفتن ۶۰۰ وظیفه، زمان مهاجرت برای روش CSO برابر با ۶۷۹.۸ واحد و برای الگوریتم ما برابر با ۶۳۱.۳ واحد اندازه‌گیری شده است. با گسترش ارزیابی به ۱۰۰۰ وظیفه، زمان مهاجرت برای الگوریتم CSO برابر با ۹۳۴.۲ واحد و برای الگوریتم ما برابر با ۸۲۱.۱ واحد است. نتایج، برتری الگوریتم ما نسبت به CSO را اثبات می‌کند و بهبود تا ۱۲٪ در زمان مهاجرت را نشان می‌دهد.

جدول ۲: ارزیابی Makespan با استفاده از بار کاری HPC2N

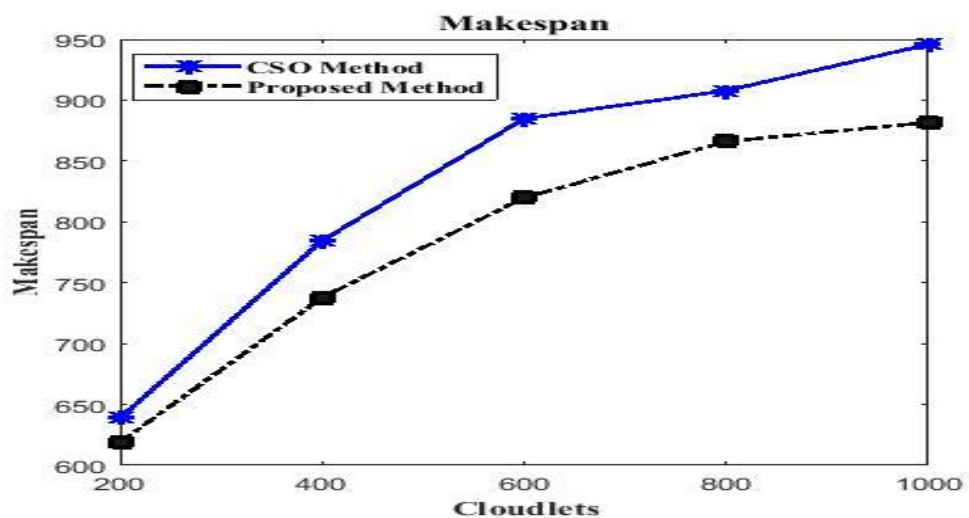
روش پیشنهادی	CSO	Cloudlets
1211.1	1251.3	200
1351.7	1358.4	400
1505.3	1547.2	600
1661.2	1742.6	800
1904.5	1985.9	1000

جدول ۳: ارزیابی Makespan با استفاده از بار کاری HPC2N

Cloudlets	CSO	روش پیشنهادی
200	641.4	620.2
400	781.2	742.6
600	878.5	819.4
800	910.5	875.6
1000	945.6	881.3



شکل ۱۳: Makespan در روش پیشنهادی و مقایسه آن به روش CSO براساس HP2CN Workload



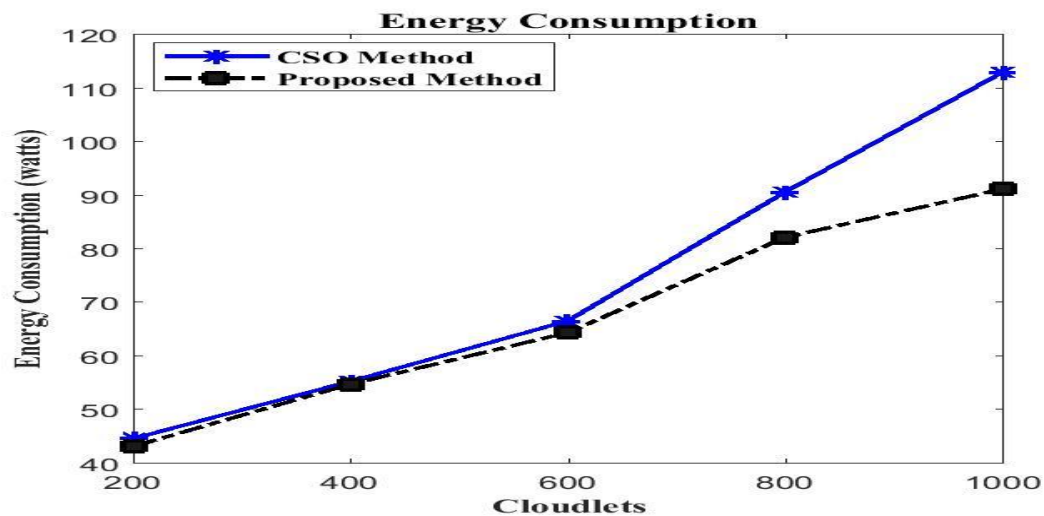
شکل ۱۴: Makespan در روش پیشنهادی و مقایسه آن با روش CSO براساس NASA Workload

جدول ۶: ارزیابی EC با استفاده از حجم کار ناسا

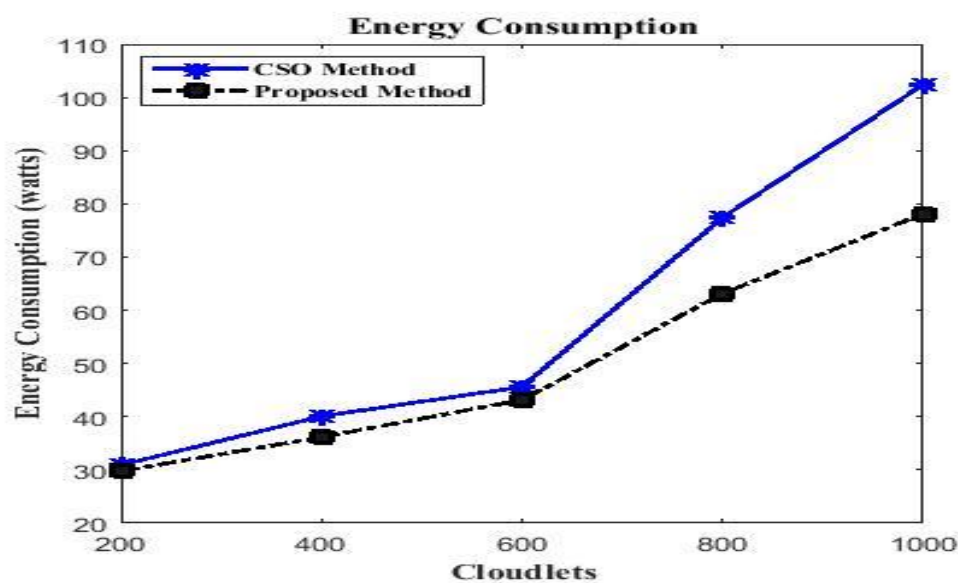
Cloudlets	CSO	روش پیشنهادی
200	32.3	30.7
400	40.4	36.2
600	45.6	43.1
800	87.3	63.4
1000	100.4	79.1

جدول ۵: ارزیابی EC با استفاده از حجم کار HPC2N

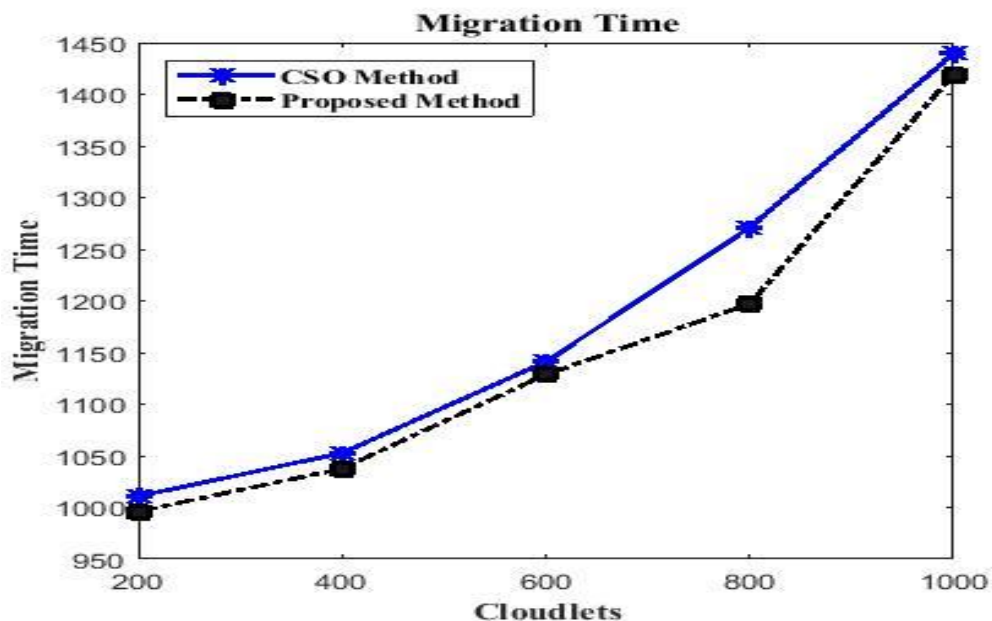
Cloudlets	CSO	روش پیشنهادی
200	1251.3	1211.1
400	1358.4	1351.7
600	1547.2	1505.3
800	1742.6	1661.2
1000	1985.9	1904.5



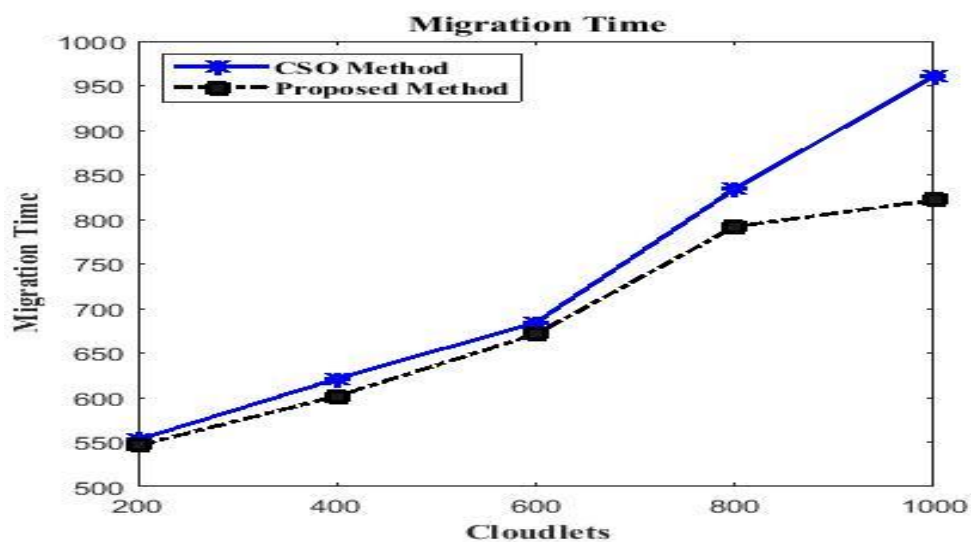
شکل ۱۵: میزان انرژی مصرفی در روش پیشنهادی و مقایسه آن به روش CSO براساس HP2CN Workload



شکل ۱۶: میزان انرژی مصرفی در روش پیشنهادی و مقایسه آن به روش CSO براساس NASA Workload



شکل ۱۷: زمان مهاجرت در روش پیشنهادی و مقایسه آن با روش CSO براساس HP2CN Workload



شکل ۱۸: زمان مهاجرت در روش پیشنهادی و مقایسه آن با روش CSO براساس NASA Workload

۵- نتیجه گیری

مطالعه حاضر رویکرد جدیدی را برای زمان بندی وظایف ابری ارائه می دهد که هدف آن به حداقل رساندن مصرف

انرژی، بهبود کیفیت زمان بندی و کاهش زمان مهاجرت است. برخلاف رویکردهای پیشین، نوآوری اصلی این تحقیق

در یکپارچه‌سازی یک مدل پیش‌بینی بار کاری مبتنی بر قوانین فرکتال با الگوریتم‌های هوش جمعی (ACO و CapSA) نهفته است. این مدل فرکتال، با توانایی خود در شناسایی الگوهای پیچیده و خودمتشابه در بارهای کاری ابری، پایه و اساس تصمیم‌گیری‌های بهینه در زمان‌بندی و مهاجرت را تشکیل می‌دهد. نتایج شبیه‌سازی به وضوح نشان داد که این رویکرد می‌تواند با مدیریت بهینه مهاجرت‌ها بر اساس پیش‌بینی‌های دقیق فرکتال و ارائه مناسب‌ترین زمان‌بندی برای وظایف کاربر، بار را متعادل کند. همچنین، با انتخاب مناسب‌ترین منابع برای تخصیص وظایف، که با بیش‌های حاصل از مدل فرکتال تقویت شده است، تحمل خطا افزایش می‌یابد. علاوه بر این، نتایج نشان داد که ترکیب زمان‌بندی مبتنی بر CapSA و به‌کارگیری مناسب‌ترین روش‌ها برای مهاجرت، که هر دو از قابلیت‌های پیش‌بینی فرکتال بهره می‌برند، مصرف انرژی منابع را به طور قابل توجهی به حداقل می‌رساند و زیرساخت مورد نیاز برای محاسبات ابری سبز را فراهم می‌کند. رویکرد پیشنهادی نسبت به رویکردهای قبلی در مورد EC، ET و تعادل بار ۱۵ تا ۲۰ درصد عملکرد بهتری دارد، که این بهبودها به طور مستقیم به دقت و کارایی مدل‌سازی فرکتال در پیش‌بینی و بهینه‌سازی سیستم نسبت داده می‌شود. محدودیت‌ها و پیشنهادها برای تحقیقات آینده به شرح زیر است.

۱. در این تحقیق، آستانه سرریز PM به صورت دستی تعیین می‌شود، اما بهتر است خودکار باشد.

۲. همچنین پارامتر هزینه محاسباتی ارزیابی نشده است، اما بهتر است آن را مطابق با سایر پارامترهای چند هدفه در نظر گرفت. از این رو، باید یک راه‌حل مصالحه بین پارامترهای متناقض حفظ شود.

منابع

- [1] I. Attiya, M. Abd Elaziz, L. Abualigah, T. N. Nguyen, and A. A. Abd El-Latif, "An improved hybrid swarm intelligence for scheduling iot application tasks in the cloud," *IEEE Transactions on Industrial Informatics*, 2022.
- [2] S. P. Singh, A. Nayyar, R. Kumar, and A. Sharma, "Fog computing: from architecture to edge computing and big data processing," *The Journal of Supercomputing*, vol. 75, no. 4, pp. 2070-2105, 2019.
- [3] S. Sefati, M. Mousavinasab, and R. Zareh Farkhady, "Load balancing in cloud computing environment using the Grey wolf optimization algorithm based on the reliability: performance evaluation," *The Journal of Supercomputing*, vol. 78, no. 1, pp. 18-42, 2022.
- [4] A. Hedhli and H. Mezni, "A survey of service placement in cloud environments ", *Journal of Grid Computing*, vol. 19, no. 3, pp. 1-32, 2021.

- [5] O. Sohaib, M. Naderpour, W. Hussain, and L. Martinez, "Cloud computing model selection for e-commerce enterprises using a new 2-tuple fuzzy linguistic decision-making method," *Computers & Industrial Engineering*, vol. 132, pp. 47-58, 2019.
- [6] M. R. Dorsala, V. Sastry, and S. Chapram, "Blockchain-based solutions for cloud computing: A survey," *Journal of Network and Computer Applications*, vol. 196, p. 103246, 2021.
- [7] S. Bharany *et al.*, "Energy efficient fault tolerance techniques in green cloud computing: A systematic survey and taxonomy," *Sustainable Energy Technologies and Assessments*, vol. 53, p. 102613, 2022.
- [8] M. Javaid, A. Haleem, R. P. Singh, S. Rab, R. Suman, and I. H. Khan, "Evolutionary Trends in Progressive Cloud Computing based Healthcare: Ideas, Enablers, and Barriers," *International Journal of Cognitive Computing in Engineering*, 2022.
- [9] A. Mohamed *et al.*, "Software-defined networks for resource allocation in cloud computing :A survey," *Computer Networks*, vol. 195, p. 108151, 2021.
- [10] S. K. Mishra *et al.*, "Energy-efficient VM-placement in cloud data center," *Sustainable computing: informatics and systems*, vol. 20, pp. 48-55, 2018.
- [11] S. Vinoth, H. L. Vemula, B. Haralayya ,P. Mamgain, M. F. Hasan, and M. Naved, "Application of cloud computing in banking and e-commerce and related security threats," *Materials Today: Proceedings*, vol. 51, pp. 2172-2175, 2022.
- [12] N. Subramanian and A. Jeyaraj, "Recent security challenges in cloud computing," *Computers & Electrical Engineering*, vol. 71, pp. 28-42, 2018.
- [13] V. Priya, C. S. Kumar, and R. Kannan, "Resource scheduling algorithm with load balancing for cloud service provisioning," *Applied Soft Computing*, vol. 76, pp. 416-424.۲۰۱۹ ,
- [14] Z. Tong, X. Deng, H. Chen, and J. Mei, "DDMTS: A novel dynamic load balancing scheduling scheme under SLA constraints in cloud computing," *Journal of Parallel and Distributed Computing*, vol. 149, pp. 138-148, 2021.
- [15] L. Ismail and A. Fardoun" ,Eats: Energy-aware tasks scheduling in cloud computing systems," *Procedia Computer Science*, vol. 83, pp. 870-877, 2016.
- [16] T. Liu, F. Chen, Y. Ma, and Y. Xie, "An energy-efficient task scheduling for mobile devices based on cloud assistant," *Future Generation Computer Systems*, vol. 61, pp. 1-12, 2016.
- [17] X. Wei, "Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-12, 2020.
- [18] M .Cheng, J. Li, P. Bogdan, and S. Nazarian, "H₂O-Cloud: A Resource and Quality of Service-Aware Task Scheduling Framework for Warehouse-Scale Data Centers," *IEEE*

Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 10, pp. ۲۰۱۹, ۲۹۳۷-۲۹۲۵ .

- [19] X. Chen *et al.*, "A WOA-based optimization approach for task scheduling in cloud computing systems," *IEEE Systems Journal*, vol. 14, no. 3, pp. 3117-3128, 2020.
- [20] Z. Tong, H. Chen, X. Deng, K. Li, and K. Li, "A scheduling scheme in the cloud computing environment using deep Q-learning," *Information Sciences*, vol. 512, pp. 1170-1191, 2020.
- [21] R. M. Abdelmoneem, A. Benslimane, and E. Shaaban, "Mobility-aware task scheduling in cloud-Fog IoT-based healthcare architectures," *Computer Networks*, vol. 179, p. 107348, 2020.
- [22] F. Ebadifard and S. M. Babamir, "Autonomic task scheduling algorithm for dynamic workloads through a load balancing technique for the cloud-computing environment," *Cluster Computing*, vol. 24, no. 2, pp. 1075-110.۲۰۲۱ .
- [23] L. Zhang, L. Zhou, and A. Salah, "Efficient scientific workflow scheduling for deadline-constrained parallel tasks in cloud computing environments," *Information Sciences*, vol. 531, pp. 31-46, 2020.
- [24] X. Zhou, G. Zhang, T. Wang, M. Zhang, X. Wang, and W. Zhang, "Makespan–Cost–Reliability-Optimized Workflow Scheduling Using Evolutionary Techniques in Clouds," *Journal of Circuits, Systems and Computers*, vol. 29, no. 10, p. 2050167, 2020.
- [25] Y. Li *et al.*, "An efficient scheduling algorithm for dataflow architecture using loop-pipelining," *Information Sciences*, vol. 547, pp. 1136-1153, 2021.
- [26] Z. Li, V. Chang, H. Hu, H. Hu, C. Li, and J. Ge, "Real-time and dynamic fault-tolerant scheduling for scientific workflows in clouds," *Information Sciences*, vol. 568, pp. 13-39, 2021.
- [27] Z. Xu, Y. Gong, Y. Zhou, Q. Bao, and W. Qian, "Enhancing kubernetes automated scheduling with deep learning and reinforcement techniques for large-scale cloud computing optimization," 2024, arXiv:2403.07905.
- [28] D. Zou, F. Wang, N. Yu, and X. Kong, "Solving many-objective optimisation problems by an improved particle swarm optimisation approach and a normalised penalty method," *International Journal of Bio-Inspired Computation*, vol. 14, no. 4, pp. 247-264, 2019.
- [29] L. Xie, Z. Hu, X. Cai, W. Zhang, and J. Chen, "Explainable recommendation based on knowledge graph and multi-objective optimization," *Complex & Intelligent Systems*, vol. 7, no. 3, pp. 1241-1252, 2021.
- [30] F. Xue and D. Wu, "NSGA-III algorithm with maximum ranking strategy for many-objective optimisation," *International Journal of Bio-Inspired Computation*, vol. 15, no. 1, pp. 14-23, 2020.

- [31] Y. Cao, L. Zhou, and F. Xue, "An improved NSGA-II with dimension perturbation and density estimation for multi-objective DV-Hop localisation algorithm," *International Journal of Bio-Inspired Computation*, vol. 17, no. 2, pp. 121-130, 2021.
- [32] J. Yuan, H.-L. Liu, F. Gu, Q. Zhang, and Z. He, "Investigating the properties of indicators and an evolutionary many-objective algorithm using promising regions," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 75-86, 2020.
- [33] M. Xu, M. Zhang, X. Cai, and G. Zhang, "Adaptive neighbourhood size adjustment in MOEA/D-DRA," *Int. J. Bio Inspired Comput.*, vol. 17, no. 1, pp. 14-23, 2021.
- [34] Z. Cui, Z. Zhang, Z. Hu, S. Geng, and J. Chen, "A many-objective optimization based intelligent high performance data processing model for cyber-physical-social systems," *IEEE Transactions on Network Science and Engineering*, 2021.
- [35] H. Yan, X. Zhu, H. Chen, H. Guo, W. Zhou, and W. Bao, "DEFT: Dynamic fault-tolerant elastic scheduling for tasks with uncertain runtime in cloud," *Information Sciences*, vol. 477, pp. 30-46, 2019.
- [36] S. Shruthi and V. Nagaveni, "A Survey on Various Parallel Power Aware Task Scheduling Algorithms for Reducing Power Consumption," 2014.
- [37] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in *Proc. 10th IEEE/ACM Int. Conf. Cluster, Cloud and Grid Computing*, 2010, pp. 826-831.
- [38] S. K. Garg, C. S. Yeo, A. Anandasivam, and R. Buyya, "Environment-conscious scheduling of HPC applications on distributed Cloud-oriented data centers," *J. Parallel Distrib. Comput.*, vol. 71, no. 6, pp. 732-749, Jun. 2011.
- [39] H. Liu, H. Jin, X. Liao, L. Hu, and C. Yu, "Live virtual machine migration via asynchronous replication and state synchronization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 12, pp. 1986-1999, Dec. 2011.
- [40] Braik, Malik, Alaa Sheta, and Heba Al-Hiary. "A novel meta-heuristic search algorithm for solving optimization problems: capuchin search algorithm." *Neural computing and applications* 33.7 (2021): 2515-2547.
- [41] H. Zhao, G. Qi, Q. Wang, J. Wang, P. Yang, and L. Qiao, "Energy-efficient task scheduling for heterogeneous cloud computing systems," in *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2019: IEEE, pp. 952-959.
- [42] B. B. Naik, D. Singh, and A. B. Samaddar, "FHCS: Hybridised optimisation for virtual machine migration and task scheduling in cloud data center," *IET Communications*, vol. 14, no. 12, pp. 1942-1948, 2020.

- [43] S. Mangalampalli, S. K. Swain, and V. K. Mangalampalli, "Multi Objective Task Scheduling in Cloud Computing Using Cat Swarm Optimization Algorithm," *Arabian Journal for Science and Engineering*, vol. 47, no. 2, pp. 1821-1830, 2022.