سال چهاردهم، شماره‌های ۵۱ و ۵۲ ، بهار و تابستان ۱۴۰۱

صص: ۱۱۳_۱۲۸

دو فصلنامه علمی

فناوری اطلاعات و ارتباطات ایران

ارائه رویکرد مبتنی بر الگوریتم تکاملی تفاضلی چندهدفه برای مسئله تخصیص منبع در محیط رایانش ابری

سعید بختیاری * **ماهان خسروشاهی** **

* استادیار گروه فتا، دانشگاه علوم انتظامی امین، تهران

** دانشجو کارشناسی ارشد مهندسی فناوری اطلاعات و ارتباطات، دانشگاه آزاد اسلامی واحد تهران مرکزی، تهران

تاریخ دریافت: ۱۴۰۰/۰۳/۲۴ تاریخ پذیرش: ۱۴۰۰/۱۲/۰۷

نوع مقاله: پژوهشی

چکیده

در سال‌های اخیر، الگوی رایانش ابری به دلیل مقیاس‌پذیری بالا، قابلیت اطمینان، اشتراک اطلاعات و هزینه پایین نسبت به ماشین‌های مجزا، بسیار مورد توجه قرارگرفته است. در محیط ابر، زمانبندی و تخصیص بهینه وظایف بر استفاده مؤثر از منابع سیستم اثر می‌گذارد. در حال حاضر روش‌های متداول برای زمانبندی در محیط رایانش ابری با استفاده از روش‌های سنتی مانند حداقل-حداقل و روش‌های فرا ابتکاری مانند الگوریتم کلونی مورچه‌ها انجام می‌شود. روش‌های فوق بر بهینه سازی یک هدف متمرکز هستند و به طور همزمان چندین هدف را برآورد نمی‌کنند. هدف اصلی این تحقیق در نظر گرفتن چندین هدف (زمان اجرای کل، توافق‌نامه سطح سرویس، مهاجرت و انرژی مصرف شده) در مراکز داده ابری با زمانبندی و تخصیص بهینه وظایف می‌باشد. در این پژوهش الگوریتم تکاملی تفاضلی چندهدفه به دلیل ویژگی‌های ساختار ساده و پارامترهای قابل تنظیم کمتر، مورد استفاده قرار می‌گیرد. در روش پیشنهادی، رویکردی جدید مبتنی بر الگوریتم تکاملی تفاضلی برای حل مسأله تخصیص در فضای ابری ارائه می‌شود که در رویکرد ارائه شده سعی می‌شود که بر اساس تابع سودمندی چندهدفه تعریف شده و در نظر گرفتن بردارهای جهش و تقاطع بتوانیم در بهبود بهره‌وری از منابع و در نظر گرفتن اهدافی چون زمان، مهاجرت و انرژی تأثیرگذار باشیم. روش پیشنهادی از طریق شبیه‌ساز کلودسیم با آزمایش بر روی حجم کار بیش از هزار ماشین مجازی بر روی داده‌های Planet Lab ارزیابی شده است. نتایج حاصل از شبیه‌سازی نشان می‌دهد که روش پیشنهادی توانسته است معیار مصرف انرژی را نسبت به الگوریتم‌های IqrMc، LrMmt و FA مقایسه شده به طور میانگین به میزان ۲۳ درصد، تعداد مهاجرت‌ها را به طور میانگین به میزان ۲۹ درصد، زمان اجرای کل را به طور میانگین به میزان ۲۹ درصد و نقص توافق‌نامه سطح سرویس را به طور میانگین به میزان ۱ درصد بهبود دهد. در این صورت استفاده از رویکرد پیشنهادی در مراکز ابری منجر به سرویس‌های بهتر و مناسب به مشتریان این مراکز در حوزه‌های مختلفی از جمله آموزش، مهندسی، صنایع تولیدی، خدماتی و... خواهد شد.

واژگان کلیدی: رایانش ابری، زمانبندی، تخصیص، الگوریتم تکاملی تفاضلی چندهدفه، مهاجرت.

نویسنده مسئول: سعید بختیاری s.b.fata@police.ir

۱. مقدمه

رایانش ابری مدلی است برای فراهم کردن دسترسی آسان بر اساس تقاضای کاربر از طریق شبکه به مجموعه‌ای از منابع رایانشی قابل تغییر و پیکربندی (مثل: شبکه‌ها، سرورها، فضای ذخیره‌سازی، برنامه‌های کاربردی و سرویس‌ها) که این دسترسی بتواند بدون کمترین نیاز به مدیریت منابع و یا دخالت مستقیم فراهم‌کننده سرویس، به سرعت فراهم شده یا آزاد گردد [۱]. رایانش ابری را می‌توان از جمله بزرگترین راه‌حل‌های مشکلات صنعت فناوری اطلاعات برشمرد. در واقع با استفاده از آن می‌توان با صرف هزینه‌های کم و بدون نگرانی از ارتقای سخت افزار و یا به‌روزرسانی نرم‌افزار مورد استفاده، از امکاناتی نظیر قدرت پردازشی و یا فضای ذخیره‌سازی سرویس دهنده‌های ابری، تنها به واسطه اتصال به اینترنت، استفاده نمود. با توجه به گسترش روزافزون استفاده از رایانش ابری، سرویس دهندگان این تکنولوژی با چالش‌های جدیدی مواجه گردیده‌اند [۲]. مسأله تخصیص منابع و زمانبندی یکی از چالش‌های مهم در رایانش ابری می‌باشد. زمانبندی به معنای واگذاری کارآمد و مناسب منابع به کارهاست. هدف اصلی آن کوتاه کردن زمان تکمیل کار، بالا بردن توان عملیاتی سیستم و ایجاد تعادل بار بر روی منابع می‌باشد. این مسئله در ابر به دلیل مقیاس بزرگ منابع، پیچیده‌تر است؛ بنابراین شناخت بهتر الگوریتم‌ها می‌تواند در انتخاب الگوریتم مناسب جهت زمانبندی و تخصیص بهینه منابع کمک زیادی کند [۱] و [۲]. تخصیص منبع و زمانبندی برای پردازش توسط منابع مناسب موجود در شبکه ابر، به عنوان یک مساله اساسی در رسیدن به کارایی بالا در سیستم رایانش ابری مطرح شده است. بر این اساس اگر به مسئله زمانبندی توجه نشود کارایی منابع ابری کاهش یافته و زمان اجرای کل افزایش می‌یابد. در حال حاضر روش‌های متداول برای زمانبندی و تخصیص منابع در محیط رایانش ابری با استفاده از روش‌های سنتی مانند روش حداقل-حداقل[۱]، روش حداکثر-حداقل[۲] و روش‌های فرا ابتکاری مانند الگوریتم کلونی مورچه‌ها[۳] و الگوریتم ژنتیک[۴] انجام می‌شود. روش‌های فوق غالباً بر بهینه سازی یک هدف متمرکز هستند و به طور همزمان چندین هدف را برآورد نمی‌کنند. در این میان برخی از الگوریتم‌های فرا ابتکاری نیاز به پارامترهای زیادی جهت تنظیم دارند و سرعت همگرایی آنها پایین است [۳]. روش‌های مختلفی مبتنی بر الگوریتم‌های فرا ابتکاری برای حل مسئله تخصیص و زمانبندی وجود دارند که هر کدام از آنها کارایی‌های مختلفی و پیچیدگی‌های متفاوتی دارند که در این مقاله الگوریتم تکاملی تفاضلی مورد مطالعه و بررسی قرار داده می‌شود.

رویکرد مبتنی بر تفاضلی دارای ویژگی‌های ساختار ساده و پارامترهای قابل تنظیم کمتر می‌باشد که برای حل انواع مسائل بهینه سازی پیچیده عددی و مهندسی منجر به نتایج رضایت بخشی شده است [۱]. از آنجایی که مسئله تخصیص منابع و زمانبندی در محیط رایانش ابری یک مسئله ان‌پی-سخت[۵] است، بدین جهت به‌کارگیری الگوریتم تکاملی تفاضلی چندهدفه در محیط رایانش ابری می‌تواند کارایی آن را به طور گسترده بهبود دهد. این الگوریتم برای غلبه بر عیب اصلی الگوریتم ژنتیک یعنی فقدان جستجوی محلی، ارائه شده است و تفاوت اصلی آن با الگوریتم ژنتیک در عملگر انتخاب می‌باشد. در رویکرد ارائه شده ابتدا سعی می‌شود که بر اساس تابع سودمندی چندهدفه تعریف شده و در نظر گرفتن بردارهای جهش و تقاطع بر اساس بهره‌وری از منابع و انرژی مصرفی قصد داریم با استفاده بهینه از منابع و بهبود بهره‌وری در کاهش زمان اجرای کل[۶] کارها، نقص توافق‌نامه سطح سرویس[۷]، تعداد مهاجرت‌ها و مصرف انرژی تأثیرگذار باشیم تا خدمات ابری مورد توجه مشتریان قرار گیرد.

۲. کارهای پیشین

رایانش ابری به عنوان یکی از انواع سیستم‌های توزیع شده که دسترسی به منابع گوناگون را از طریق سرویس‌ها بر روی بستر اینترنت فراهم می‌کند روز به روز در حال گسترش می‌باشد. از آنجایی که وظیفه ابر پاسخگویی به حجم بالای وظایف دریافتی است زمانبندی وظایف در رایانش ابری، مسأله بسیار مهمی می‌باشد که سعی دارد یک زمانبندی بهینه برای اجرای وظایف مشخص نماید. روش‌های بسیاری به ویژه روش‌های اکتشافی و تکاملی برای این مسأله ارائه گردیده است [۴]. که در ادامه به بررسی برخی از این پژوهش‌های اخیر می‌پردازیم.

در مقاله [۵] از مزایای روش مقیاس گذاری فرکانس و ولتاژ پویا[۸] همراه با رویکرد تجمیع[۹] برای ارائه دادن یک روش جدید چند معیاره فازی و راه حل مدیریت منبع واقعی استفاده شده است. ارزیابی گسترده الگوریتم‌های پیشنهادی با استفاده از شبیه‌ساز کلودسیم، کاهش‌های قابل توجهی در انرژی مصرفی، نقص توافق‌نامه سطح سرویس و تعداد مهاجرت‌ها در مقایسه با بهترین فناوری روز موجود نشان می‌دهد. این مقاله به بهبود زمان اجرای کل در محیط ابر توجه نکرده است.

مقاله [۶] روشی بر پایه الگوریتم زمانبندی انرژی آگاه[۱۰] برای تخصیص ماشین‌های مجازی ارائه نموده است. هدف اصلی این پژوهش کاهش مصرف انرژی و کاهش زمان پاسخ ماشین‌های مجازی می‌باشد. این مقاله همچنین وظایفی که بر روی ماشین‌های مجازی

---

۱ Min-Min
۲ Max-Min
۳ Ant Colony Optimization (ACO)
۴ Genetic
۵ NP-Hard
۶ Total Execution time
۷ Service-level agreement (SLA)
۸ Dynamic voltage and frequency scaling (DVFS)
۹ Consolidation
۱۰ Energy-Aware

اجرا می‌شوند را به گونه‌ای برای اجرا، اولویت بندی می‌کند تا این وظایف در کمترین زمان ممکن اجرا گردند. این مقاله به بهبود زمان اجرای وظیفه و زمان واقعی در محیط ابر توجه نکرده است.

تجمیع پویای ماشین مجازی معیار مهمی در بهبود بهره‌وری منابع می‌باشد. در مقاله [۷] یک سیاست تخصیص ماشین مجازی جدید به‌نام SABFD[۱۱] و یک سیاست مهاجرت جدید برای انتخاب ماشین مجازی به‌نام انتخاب ماشین مجازی برای مهاجرت مبتنی بر بهره پردازنده پیشنهاد شده است. نتایج تجربی آزمایشات، بهبود انرژی مصرفی و کارایی را در استفاده از این دو سیاست نسبت به برنامه‌های تجمیع پویای ماشین مجازی موجود نشان می‌دهد. از معایب این مقاله می‌توان به عدم توجه به زمانبندی و تخصیص منابع نام برد.

در مقاله [۸] یک الگوریتم بهینه سازی ترکیبی مبتنی بر الگوریتم ژنتیک و تکاملی تفاضلی ارائه شده است. با توجه به اینکه فرایند زمانبندی وظایف در محیط ابر یک ویژگی طبیعت پویا دارد بر این اساس محدودیت تابع هدف تنها از این جهت نمی‌تواند خواسته‌های کاربران را برآورده نماید. با توجه به همین مشکل یک الگوریتم چندهدفه مبتنی بر الگوریتم ژنتیک و الگوریتم تکاملی تفاضلی در این مقاله ارائه می‌شود که در الگوریتم پیشنهادی زمان کل، هزینه و تعادل بار به طور همزمان بررسی می‌گردد. در فاز جمعیت اولیه و ترکیب مجدد تنوع جمعیت اولیه با معرفی فاکتورهای مختلف فرد افزایش می‌یابد که می‌تواند مانع عبور افراد مشابه و رعایت قوانین درونی نسل‌های طبیعی شود. معرفی الگوریتم تکاملی تفاضلی در مرحله جهش الگوریتم ژنتیک می‌تواند نه تنها به مزایای قابلیت جستجوی سراسری الگوریتم ژنتیک منجر شود بلکه باعث سرعت بخشیدن به الگوریتم برای بهینه سازی راه حل با استفاده از قابلیت‌های جستجوی محلی و سرعت همگرایی سریع الگوریتم تکاملی تفاضلی می‌شود. نتایج شبیه‌سازی در کلودسیم نشان می‌دهد که این الگوریتم می‌تواند دو الگوریتم ژنتیک و تفاضلی را با توجه به کیفیت خدمات و تعادل بار ماشین مجازی در شرایط مشابه بهینه سازد و یک الگوریتم کارآمد برای زمانبندی وظایف در محیط ابر باشد. روش پیشنهادی را می‌توان تنها در محیط محاسبات پایدار استفاده کرد. درحالیکه در دنیای واقعی، استفاده از منابع محاسباتی به‌صورت تمام وقت مفید نیستند. بنابراین بهتر است اهداف دیگری مانند استحکام، قابلیت اطمینان و امنیت علاوه بر اهداف مطالعه شده برای کارهای بعدی مورد توجه قرار گیرد.

مقاله [۹] یک رویکرد ترکیبی از الگوریتم زمانبندی وظایف را که ترکیبی از ویژگی‌های مطلوب دو الگوریتم ابتکاری الهام گرفته از زیست شناسی می‌باشد، ارائه کرده است. این رویکرد ترکیبی از الگوریتم‌های ژنتیک و الگوریتم‌های جستجوی باکتری[۱۲] در رایانش ابری است. هدف اصلی این مقاله حداقل‌سازی زمان پاسخ و کاهش مصرف انرژی است. در این رویکرد به مساله سربار محاسباتی توجهی نشده است. در این مقاله به معیارهای بهبود کیفیت سرویس و کاهش تعداد مهاجرت‌ها به منظور بهبود مناسب‌تر کاهش مصرف انرژی توجهی نکرده است.

در مقاله [۱۰] به معرفی مکانیسم داوری در استراتژی انتخاب که می‌تواند زمان اجرای الگوریتم تکاملی تفاضلی را کاهش دهد، پرداخته شده است. در این مقاله به طور مؤثر کاستی‌های الگوریتم تکامل دیفرانسیل استاندارد را با سرعت همگرایی آهسته بهبود بخشیده است. در این مقاله بر بهبود الگوریتم تکاملی چندهدفه با هدف کاهش زمان متمرکز شده است. در این مقاله به معیارهای بهبود انرژی و کیفیت سرویس توجهی نشده است.

در مقاله [۱۱] یک رویکرد تکاملی جدید و مؤثر، برای تخصیص ماشین‌های مجازی که می‌تواند بهره‌وری انرژی در مرکز داده ابری را با در نظر گرفتن ماشین‌های مجازی رزرو شده بیشتر به حداکثر رساند معرفی شده است. این رویکرد می‌تواند دسترسی سریع به یک راه حل تخصیص بهینه برای یک دسته از ماشین‌های مجازی رزرو شده را فراهم کند و در عین حال ماشین‌های مجازی بیشتری را با ماشین‌های فیزیکی کمتری تجمیع نماید تا بهینه‌سازی انرژی بهتری را در مقایسه با روش‌های موجود ایجاد نماید. در این مقاله به کاهش زمان اجرا و تعداد مهاجرت‌ها توجهی نشده است.

در مقاله [۱۲] یک رویکرد چندهدفه برای زمانبندی جریان کاری در محیط‌های ابری پیشنهاد شده است. در این تحقیق نویسنده یک الگوریتم ژنتیکی چند منظوره جدید برای زمانبندی جریان کار در محیط ابر ارائه داده است. نتایج شبیه سازی نشان می‌دهد که الگوریتم پیشنهادی نه تنها از نظر بودجه، مهلت و انرژی بهبود یافته بلکه در استفاده از منابع ابر نیز در مقایسه با الگوریتم‌های رقابتی بهبود یافته است. این مقاله به اهداف دیگری مانند سطح امنیتی، اعتماد و رضایت کاربر هنگام زمانبندی توجه نکرده است. همچنین باید تحمل خطا در هنگام ترسیم وظایف گردش کار به منابع ابری مورد توجه قرار گیرد.

مقاله [۱۳] یک الگوریتم اکتشافی چندوجهی هیبریدی جدید مبتنی بر الگوریتم ژنتیکی[۱۳] (NSGA-II) و الگوریتم جستجوی گرانش[۱۴] (GSA) برای تسهیل انتخاب ماشین‌های مجازی برای زمانبندی وظایف ارائه کرده است. نتایج شبیه سازی نشان می‌دهد که الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌های زمانبندی چندهدفه عملکرد بهتری داشته و اهداف تعیین شده را برآورده می‌کند. در مسئله زمانبندی پویا عوامل دیگری مانند دما تأثیر دارند و سربار انرژی باید با جزئیات مورد بررسی قرار گیرد.

[۱۱] Space Aware Best Fit Decreasing (SABFD)
[۱۲] Bacterial Foraging (BF) Algorithms
[۱۳] Non-dominated Sorting Genetic Algorithm-II
[۱۴] Gravitational Search Algorithm

در مقاله [۱۴] یک الگوریتم زمانبندی را که از رفتار انگلی پرنده فاخته تقلید می‌نماید، پیشنهاد شده است. فاخته‌ها با بهره‌گیری از لانه‌های دیگر پرندگان، تخم‌های مشابه با خود را تکثیر می‌کنند. الگوریتم زمانبندی فاخته پیشنهاد شده را تحت تعداد مختلفی از گره ها و درصد کارها با اولویت بالا ارزیابی می‌کند. نتایج تجربی نشان می‌دهد که الگوریتم پیشنهادی از نظر معیارهایی چون میانگین استفاده از پردازنده و متوسط زمان چرخش برای هر نوع کار برتری داشته است.

تخصیص، زمانبندی و کارایی انرژی موضوعات مهمی در سیستم‌های ابری می‌باشند. مقاله [۱۵] به تخصیص و زمانبندی جهت حداقل کردن تعداد سرورهای فعال پرداخته است. در این مقاله از الگوریتم کرم شب تاب جهت فرآیند تخصیص استفاده شده است. این الگوریتم متعلق به الگوریتم‌های تصادفی می‌باشد بدین معنا که یک نوع جستجوی تصادفی برای رسیدن به مجموعه‌ای از راه حل‌ها به کار برده می‌شود. مشخصه اصلی کرم‌های شب تاب نور چشمک زن آن‌ها است. کرم‌های شب تاب به سمت کرمی می‌روند که پرنورتر است. نتایج شبیه‌سازی نشان دهنده این است که روش ارائه شده در زمان اجرا بهبود داشته است. از معایب این مقاله می‌توان به عدم توجه به معیارهای انرژی و بهبود بهره‌وری از منابع نام برد.

رایانش ابری در دهه اخیر رشد سریعی داشته است، از این روی نگرانی فزاینده‌ای در مورد انرژی مورد نیاز مراکز داده ابر وجود دارد. یک روش برای حل این مساله، تجمیع پویای ماشین مجازی[۱۵] (DVMC) است که در آن ماشین‌های مجازی (میزهای مجازی) در تعداد کمتری میزبان قرار می‌گیرند، در حالی که میزبانانی که ماشین مجازی ندارند در حالت خواب قرار داده می‌شوند. DVMC به طور سنتی با استراتژی‌هایی که بر اساس توپولوژی متمرکز ساخته می‌شوند، حل می‌شود. یکی از معایب این راه‌حل‌ها این است که آن ها در مراکز داده بسیار بزرگ مقیاس خوبی ندارند زیرا گره مدیر به یک تنگنا و یک نقطه خرابی تبدیل می‌شود. بر این اساس در مقاله [۱۶] قراردادهای شایعه پراکنی[۱۶] (GC) را که یک چارچوب جدید چند عاملی برای تهیه استراتژی‌های همکارانه غیر متمرکز است پیشنهاد شده است. GC از پروتکل شایعه[۱۷] و پروتکل قرارداد خالص[۱۸] الهام گرفته است. با استفاده ازGC، یک استراتژی GC مبتنی بر DVMC توسعه داده شده است و آن با دو استراتژی محبوب Sercon که متمرکز است و ecoCloud که استراتژی توزیع شده می‌باشد، مقایسه شده است. در این روش کاهش مصرف انرژی در نظر گرفته شده اما به مساله سربار محاسباتی توجهی نشده است.

محاسبات مبتنی بر ابر، علی‌رغم مزایای بسیار زیاد آن، تأثیرات نامناسبی بر محیط زیست نیز دارد. انتشار گازهای گلخانه‌ای و انرژی مصرفی توسط مراکز داده ابر، مهمترین مسأله‌ای است که نیاز به توجه جدی دارد. تجمیع ماشین مجازی یک روش اساسی برای صرفه‌جویی در مصرف انرژی و استفاده بهینه از منابع می‌باشد. با این حال، رویکردهای تجمیع ماشین مجازی موجود بدون در نظر گرفتن عوامل کیفیت سرویس[۱۹] به طور تهاجمی باعث کاهش مصرف انرژی می‌شوند. برخلاف راه‌حل‌های موجود، راه حل ارائه شده در مقاله [۱۷]، یک چارچوب عمومی است زیرا، هم برای پردازنده و هم برای کارهای فشرده ورودی / خروجی کار می‌کند. اثربخشی چارچوب پیشنهادی با استفاده از مجموعه داده واقعی پلتفرمPlanet Lab و CloudSim نشان داده شده‌است. راه‌حل پیشنهادی را می‌توان در مراکز داده ابری استفاده کرد تا محاسبات کارآمد انرژی را فعال نماید. این مقاله به معیارهای کیفیت سرویس و کاهش مهاجرت توجهی نکرده است.

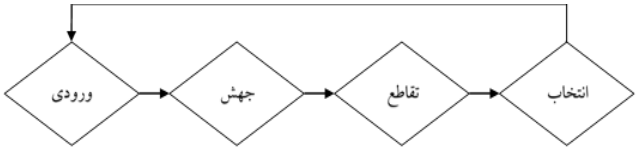
آگاهی از انرژی، چالش بزرگی برای زیرساخت‌های رایانش ابری و توسعه مراکز داده نسل بعدی است. تجمیع ماشین‌های مجازی یکی از تکنیک‌هایی است که می‌تواند برای کاهش هزینه‌های مربوط به انرژی و مسائل پایداری زیست محیطی مراکز داده استفاده گردد. در زمان‌های اخیر ثابت شده است که رویکردهای یادگیری هوشمند برای مدیریت منابع در مراکز داده ابری مؤثر می‌باشند. در مقاله [۱۸] به بررسی کاربرد الگوریتم‌های یادگیری تقویتی[۲۰] (RL) برای مساله تجمیع ماشین مجازی، که ظرفیت آن‌ها را برای بهینه‌سازی توزیع ماشین‌های مجازی در مرکز داده برای بهبود منابع نشان می‌دهد، پرداخته شده است. تعیین سیاست‌های کارآمد در محیط‌های پویا می‌تواند کار دشواری باشد، با این حال، روش پیشنهادی RL به دلیل توانایی ذاتی استدلال در عدم اطمینان در صورت عدم دانش کامل، رفتار بهینه را می‌آموزد. با استفاده از داده‌های بار کاری واقعی، یک تحلیل مقایسه‌ای از الگوریتم‌های محبوب RL از جملهSARSA و Q-learning ارائه داده شده است. نتایج تجربی نشان می‌دهند چگونه این رویکرد ۲۵٪ بهره‌وری انرژی را بهبود می‌بخشد در حالی که ۶۳٪ نقض خدمات را نسبت به الگوریتم اکتشافی محبوب انرژی آگاه کاهش می‌دهد. در این مقاله به معیارهای زمان و مهاجرت توجهی نشده است.

[۱۵] Dynamic Virtual Machine Consolidation (DVMC)
[۱۶] Gossip Contracts (GC)
[۱۷] Gossip
[۱۸] Contract Net Protocol (CNP)
[۱۹] Quality of Service (QoS)
[۲۰] Reinforcement Learning

۳. روش پیشنهادی

۳. ۱ کلیت روش پیشنهادی

الگوریتم تکاملی تفاضلی یک الگوریتم جدید اکتشافی است که دارای ویژگی‌های ساده و پارامترهای قابل تنظیم کمتر می‌باشد که برای حل انواع مسائل بهینه سازی پیچیده عددی و مهندسی منجر به نتایج رضایت بخشی شده است. از آنجائی‌که مسئله زمانبندی و تخصیص در محیط ابری یک مسئله پیچیده سخت است، بدین جهت کاربرد الگوریتم تکاملی تفاضلی در تخصیص منابع در محیط ابری می‌تواند کارایی را به طور قابل توجهی بهبود دهد. همچنین مسئله مصرف انرژی در سیستم‌های ابری مسئله مهمی است و باید منابع به شیوه‌ای مناسب مدیریت شوند و به گونه‌ای ماشین‌های مجازی به منابع محاسباتی تخصیص داده شوند که منابع دچار پرباری نشوند.
با توجه به ماهیت الگوریتم تکاملی تفاضلی که به‌دنبال جواب بهینه هستند برای انتخاب میزبان مناسب برای تخصیص و جای‌دهی مناسب با هدف بهبود فرایند زمانبندی و کاهش مصرف انرژی از الگوریتم تکاملی تفاضلی استفاده می‌گردد. با توجه به این تعاریف روند کلی الگوریتم تکاملی تفاضلی در شکل ۱ نمایش داده شده است. این الگوریتم از چهار مرحله تشکیل شده است که در شکل زیر نمایش داده شده است.



شکل ۱. روند کلی الگوریتم تکاملی تفاضلی

در ادامه جزئیات روش پیشنهادی بیان شده است.

۳. ۱. ۱ گام اول: ورودی‌های الگوریتم پیشنهادی

با توجه به شکل بالا، این الگوریتم دارای چهار مرحله اصلی شامل ورودی، جهش، تقاطع و انتخاب می‌باشد. برای حل مسئله پیشنهادی ابتدا به گام اول، ورودی‌ها در روش پیشنهادی می‌پردازیم. در گام اول ماشین‌های مجازی و میزبان‌های فیزیکی به عنوان مؤلفه‌های ورودی مسئله در نظر گرفته می‌شوند. بعد از اینکه ورودی‌های مسئله مشخص شد، بردارهایی از آنها تشکیل می‌گردد. بردارهای مسئله به تعداد ماشین‌های مجازی و میزبان‌های فیزیکی به عنوان جمعیت اولیه در نظر گرفته می‌شوند.

۳. ۱. ۲ گام دوم: جهش در الگوریتم پیشنهادی

با توجه به بردارهای ایجاد شده در مرحله قبل، مشخصات ماشین‌های مجازی و مشخصات میزبان‌های فیزیکی به عنوان جمعیت اولیه برای ورودی الگوریتم تکاملی تفاضلی تعیین می‌گردد. با توجه به ورودی‌های ایجاد شده در مرحله قبل، به‌دنبال میزبان فیزیکی می‌گردیم که بتواند بهترین تخصیص برای ماشین‌های مجازی و درخواست‌ها به لحاظ منابع آزاد داشته باشد. هر چه پردازنده و حافظه آزاد بیشتر باشد، احتمال پربار شدن سرور و مهاجرت اضافی که در افزایش زمان و مصرف انرژی تأثیرگذار است، کاهش پیدا می‌کند. در نتیجه برای در نظر گرفتن عملگر جهش از آنجا که این بردار در ابتدا بدترین حالت در نظر گرفته می‌شود، در نتیجه بردار جهش با مقدار موقت تابع هدف بیشتر مقداردهی می‌شود.

۳. ۱. ۳ گام سوم: تقاطع در الگوریتم پیشنهادی

در هر مرحله از الگوریتم تکاملی تفاضلی بردار تقاطع محاسبه می‌شود. بردار تقاطع بر اساس تابع هدف سنجیده می‌شود و اگر مقدار تابع هدف کمتر از مقدار قبلی باشد، آنگاه در مرحله انتخاب، این مقدار جایگزین مقدار قبلی می‌گردد. این عملیات بر اساس تعداد منابع محاسباتی که در نظر گرفته شده است تکرار می‌شود تا به بهترین جواب برای به دست آوردن بهترین منبع محاسباتی دست‌یابد.

۳. ۱. ۴ گام چهارم: انتخاب در الگوریتم پیشنهادی

در گام سوم، چنانچه مقدار تابع هدف محاسبه شده توسط بردار تقاطع کمتر از مقدار آن در بردار جهش باشد، آنگاه بردار تقاطع جایگزین بردار جهش از نسل قبل می‌گردد. این مرحله انتخاب نام دارد و توسط تابع هدف مشخص می‌گردد. تمامی مراحل جهش، تقاطع و انتخاب تکرار می‌گردد و در پایان بهترین جواب جایگزین بردار قبلی می‌شود. برای این منظور فرمول‌های زیر جهت تقاطع، توسط تابع هدف برای انتخاب بهترین منبع محاسباتی جهت تخصیص در نظر گرفته شده است.

$$\text{fitness} = \text{MIN}\left(\frac{\text{RequestedCPU}_{\text{VM}_i}}{\text{AllocatedCPU}_{\text{PM}_j}} + \frac{\text{RequestedRAM}_{\text{VM}_i}}{\text{AllocatedRAM}_{\text{PM}_j}} + w_{\text{PM}_j}\right) \quad (1)$$

$$i = ۱،۲..., N, j = ۱،۲..., M$$

در نتیجه رابطه (۱) برای تابع هدف در نظر گرفته شده است.
در رابطه (۱)، $\text{RequestedCPU}_{\text{VM}_i}$ میزان پردازنده درخواست شده توسط ماشین مجازی i ام، $\text{AllocatedCPU}_{\text{PM}_j}$ میزان پردازنده اختصاص داده شده به میزبان فیزیکی j ام، w_{PM_i} میزان بار استفاده شده توسط میزبان فیزیکی j ام، $\text{RequestedRAM}_{\text{VM}_i}$ میزان حافظه درخواست شده توسط ماشین مجازی i ام، $\text{AllocatedRAM}_{\text{PM}_j}$ میزان حافظه اختصاص داده شده توسط میزبان فیزیکی j ام می‌باشد.
بر اساس تابع هدف ارائه شده میزبانی که منابع بیشتری در اختیار ماشین مجازی قرار دهد میزبان مناسب‌تری برای جای‌دهی ماشین مجازی می‌باشد. در نتیجه بر اساس تابع هدف تعریف شده میزبانی که پردازنده آزاد و حافظه بیشتری در اختیار ماشین مجازی قرار دهد میزبان مناسب‌تری برای تخصیص می‌باشد. با توجه به تابع هدف ارائه شده در واقع میزبانی که منابع آزاد بیشتری دارد مناسب‌تر است و همچنین میزان حافظه بر روی زمان مهاجرت تأثیر دارد بر همین

اساس در تابع هدف، استفاده از حافظه در نظر گرفته شده است. تابع هدف الگوریتم تکاملی تفاضلی با توجه به شرایط مسئله تکرار می‌شود تا بهترین جواب بهینه به دست آید و در نهایت میزبان فیزیکی که پردازنده آزاد و حافظه بیشتر دارد، برای تخصیص در نظر گرفته می‌شود. از آنجا که پردازنده بیشترین تأثیر را در مصرف انرژی دارد با در نظر گرفتن پردازنده در تابع هدف پیشنهادی باعث می‌شود میزبانی که پردازنده آزاد بیشتری دارد جهت تخصیص انتخاب شود در واقع این میزبان احتمال پربار شدن آن کمتر است و از مهاجرت اضافی که در افزایش زمان و انرژی مصرفی تأثیرگذار است جلوگیری می‌کند و همچنین از تعداد سرورهای فعال کمتری استفاده خواهد شد در نتیجه کاهش سرورها باعث کاهش مصرف انرژی نیز می‌شود.

۳. ۲ تجمیع پویای ماشین مجازی

تجمیع پویای ماشین مجازی، ماشین‌های مجازی را در تعداد سرورهای کمتری قرار می‌دهد تا از منابع بهینه استفاده کند و مصرف انرژی را کاهش دهد. فرایند تجمیع پویای ماشین‌های مجازی بعد از تخصیص و جهت بهینه‌سازی آن صورت می‌گیرد که در حالت کلی به سه مرحله تقسیم می‌شود، که شامل:

مرحله اول: شناسایی میزبان فیزیکی پربار، شناسایی میزبان‌های فیزیکی که دچار پرباری بیش از حد شده‌اند که نیاز به مهاجرت یک یا چند ماشین مجازی از این میزبان‌های فیزیکی برای جلوگیری از نقض کیفیت سرویس وجود دارد.

مرحله دوم: انتخاب ماشین مجازی، انتخاب ماشین مجازی مناسب جهت مهاجرت از منبعی که دچار پرباری شده است.

مرحله سوم: انتخاب میزبان فیزیکی، انتخاب مقصد مناسب برای جای‌دهی ماشین‌های مجازی مهاجر.

۳. ۲. ۱ گام اول: شناسایی میزبان فیزیکی پربار

در مقاله [۱۹]، مصطفی و همکاران چهار مکانیزم آستانه پویا جهت تشخیص سرور پربار پیشنهاد کرده‌اند که شامل متوسط انحراف مطلق[۲۱] (MAD)، رگرسیون محلی[۲۲] (LR) و محدوده بین چارک[۲۳] (IQR) می‌باشد. در روش پیشنهادی جهت شناسایی سرورهای پربار برای جلوگیری از پرباری و نقض کیفیت سرویس از روش MAD استفاده شده است. ما MAD را به دلیل اینکه در مقایسه با مدل‌های خارجی مقاوم است، انتخاب کردیم. علاوه بر این، MAD از دانش قبلی برای ایجاد آستانه جدید استفاده می‌کند.

سیاست MAD یک تخمین‌گر قوی از واریانس نمونه یا انحراف استاندارد برای یک مجموعه داده تک متغیری $X_1, X_2, \ldots, X_n$ است.MAD به‌عنوان متوسط انحراف مطلق از میانگین داده‌ها تعریف می‌شود [۱۹]. در رابطه (۲) این سیاست بیان شده است.

$$\mathrm{MAD} = \mathrm{median}_i(|X_i - \mathrm{median}_j(X_j)|) \qquad (۲)$$

MAD با انحراف میانگین داده‌ها شروع می‌شود و متوسط مقدار مطلق آنها است. بر این اساس حد بالای آستانه به صورت رابطه (۳) تعریف می‌شود که $s \in R^+$ و امنیت روش را تنظیم می‌کند.

$$T_u = 1 - s \cdot \mathrm{MAD} \qquad (۳)$$

۳. ۲. ۲ گام دوم: انتخاب ماشین مجازی

برای سیاست مهاجرت و انتخاب ماشین مجازی از میزبان پربار از سیاست حداقل زمان مهاجرت[۲۴] (MMT) استفاده می‌کنیم. این سیاست ماشین مجازی که، به کمترین زمان برای تکمیل یک مهاجرت نسبت به دیگر ماشین‌های مجازی تخصیص یافته به میزبان فیزیکی دارد، برای مهاجرت انتخاب می‌کند. زمان مهاجرت با مقدار حافظه استفاده شده توسط ماشین مجازی تقسیم بر پهنای باند شبکه در دسترس برای میزبان فیزیکی j ام محاسبه می‌شود. V_j مجموعه‌ای از ماشین‌های مجازی می‌باشد که اخیراً به میزبان فیزیکی j ام تخصیص یافته است. سیاست MMT ماشین مجازی که در آن υ شرایط فرمول ارائه شده در رابطه (۴) را برآورده کند، پیدا می‌کند [۱۹].

$$\upsilon \in V_j | \forall \alpha \in V_j . \frac{\mathrm{RAM_u}(\upsilon)}{\mathrm{NET_j}} \leq \frac{\mathrm{RAM_u}(\alpha)}{\mathrm{NET_j}} \qquad (٤)$$

$\mathrm{RAM_u}(\alpha)$ مقدار RAM استفاده شده اخیر توسط ماشین مجازی α می‌باشد. $\mathrm{NET_j}$ پهنای باند شبکه در دسترس برای میزبان فیزیکی j ام می‌باشد.

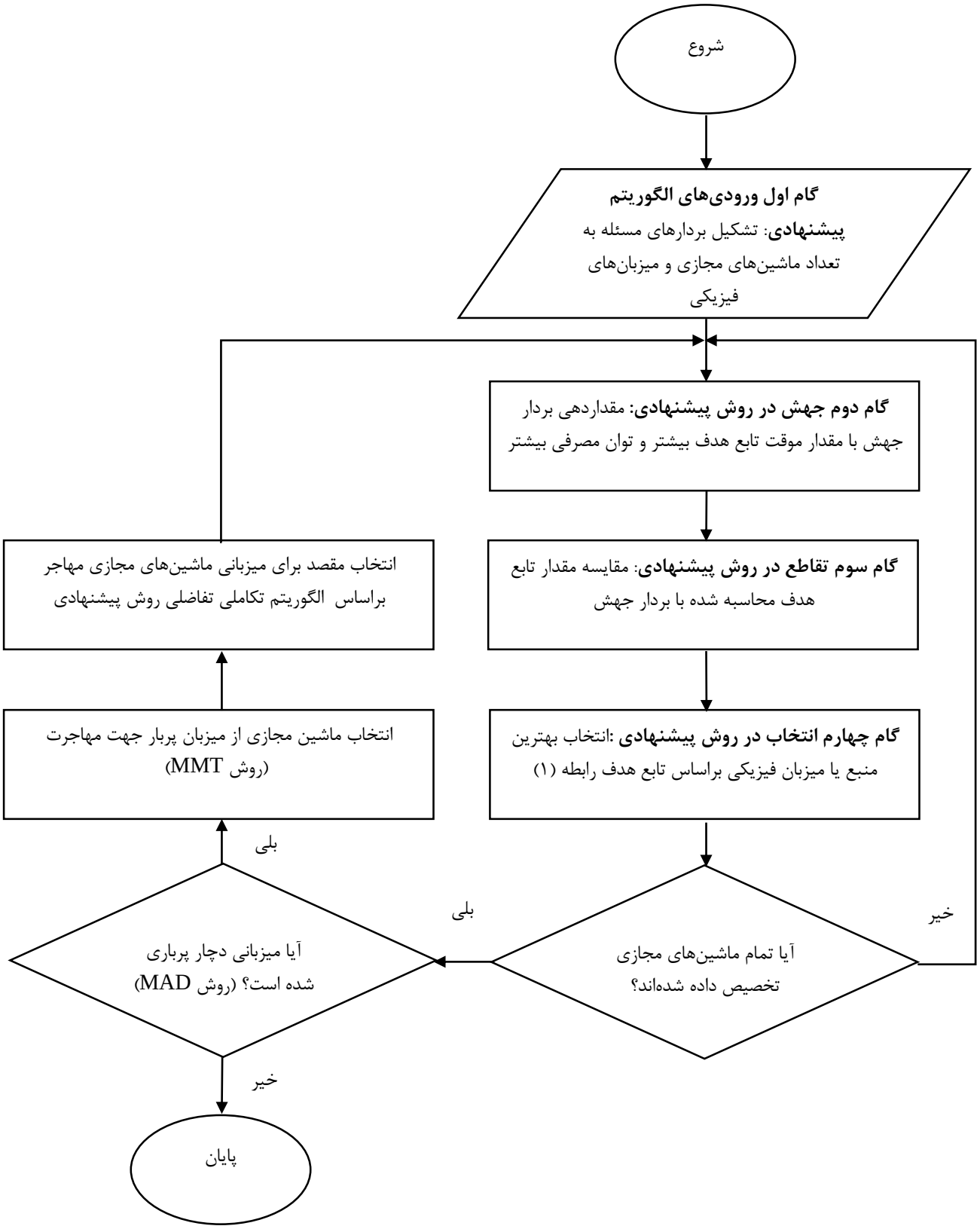
۳. ۲. ۳ گام سوم: انتخاب میزبان فیزیکی

در نهایت برای انتخاب مقصد برای میزبانی ماشین‌های مجازی مهاجر از سیاست پیشنهادی جهت فرایند تخصیص و استفاده از الگوریتم تکاملی تفاضلی پیشنهاد شده استفاده می‌شود.

[۲۱] Medium Absolute Deviation (MAD)
[۲۲] Local Regression (LR)
[۲۳] Interquartile Range (IQR)
[۲۴] Minimum Migration Time (MMT)

۳. ۳ فلوچارت روش پیشنهادی

برای این منظور در شکل ۲ مراحل الگوریتم بیان شده است.



شکل۲. فلوچارت روش پیشنهادی

۳. ۴ معیارهای ارزیابی روش پیشنهادی

برای این منظور معیارهایی که قصد داریم روش پیشنهادی خود را با آن مقایسه کنیم شامل انرژی مصرفی میزبان‌های فیزیکی، تعداد مهاجرت‌ها، نقض توافق‌نامه سطح سرویس و زمان اجرای کل می‌باشد. زمان اجرای کل موضوع مهمی است که با تخصیص، زمانبندی مناسب و استفاده بهینه از منابع می‌توانیم در بهبود این معیار تأثیرگذار باشیم. همچنین انرژی مصرفی میزبان‌های فیزیکی با تخصیص، جای‌دهی مناسب و استفاده بهینه از منابع قابل بهبود است. انرژی مصرف شده مراکز داده ابری بر اساس بهره‌وری از پردازنده در نظر گرفته می‌شود. با توجه به اینکه استفاده از پردازنده نقش مهمی در انرژی مصرفی در مقایسه با منابع دیگر دارد. یک منبع فیزیکی بیکار تقریباً هفتاد درصد نسبت به میزبانی که از تمام بهره پردازنده خود استفاده می‌کند، انرژی مصرف می‌کند [۱۸]. در نتیجه معیاری که برای انرژی مصرفی در نظر گرفته شده است، هم مربوط به زمانی که یک سرور بیکار است و هم مربوط به زمانی است که از پردازنده خود حداکثر استفاده را می‌کند. در نتیجه برای این منظور مصطفی و همکاران [۱۹] در سال ۲۰۱۸ فرمولی بر اساس بهره پردازنده برای اندازه‌گیری میزان انرژی مصرفی منابع فیزیکی ارائه کرده‌اند که در رابطه (۵) بیان شده است.

$$P(u) = ۰,۷ \times P_{max} + ۰,۳ \times P_{max} \times U \qquad (۵)$$

بر اساس رابطه (۵)، P_{max} حداکثر انرژی مصرف شده هنگامی که یک منبع فیزیکی به طور کامل مورد استفاده قرار می‌گیرد می‌باشد. این معیار در آزمایشات مقدار ۲۵۰ وات در نظر گرفته می‌شود که یک مقدار مناسب برای منابع فیزیکی می‌باشد. U بهره‌وری از پردازنده است. با توجه به اینکه بهره‌وری از پردازنده با گذشت زمان تغییر می‌کند این معیار به صورت انتگرال بیان می‌شود. برای این منظور، بهره‌وری از پردازنده را به صورت تابعی از زمان بیان کرده‌اند که این فرمول در رابطه (۶) ارائه شده است [۱۹].

$$E_i = \int_{t_۰}^{t_۱} P\big(u(t_i)\big)dt \qquad (۶)$$

با توجه به رابطه ارائه شده مصرف کل انرژی توسط میزبان فیزیکی i ام به صورت انتگرال تابع مصرف انرژی روی یک دوره زمانی $t_۰$ تا $t_۱$ محاسبه می‌شود. $u(t_i)$ میزان استفاده از پردازنده میزبان فیزیکی i ام را به صورت تابعی از زمان نشان می‌دهد.

در نتیجه کل مصرف انرژی مراکز داده ابری را می‌توان به عنوان مجموع عملکرد توان مصرفی همه میزبان‌ها تعریف کرد، همانطور که در رابطه (۷) نشان داده شده است.

$$E_{DataCenter} = \sum_{i=۰}^{N} E_i \qquad (۷)$$

بر اساس رابطه (۷) N تعداد میزبان‌های فیزیکی و E_i توان مصرفی میزبان فیزیکی i ام می‌باشد.

در محیط ابری برای در نظر گرفتن کیفیت سرویس، معیاری به نام توافق‌نامه سطح سرویس (SLA) در نظر گرفته شده است. نقض توافق‌نامه سطح سرویس در محیط محاسبات ابری بسیار نامطلوب است و هزینه‌های سنگینی دارد. برای این منظور دو معیار برای اندازه‌گیری سطح نقض توافق‌نامه سطح سرویس در نظر گرفته می‌شود [۱۹].

- SLATAH[۲۵] (نقض توافق‌نامه سطح سرویس میزبان فعال): زمانی که ماشین‌های فیزیکی فعال هستند و بهره ۱۰۰٪ پردازنده را تجربه می‌کنند و دچار پرباری می‌شوند.
- PDM[۲۶] (نقض کارایی با توجه به مهاجرت): کاهش کارایی کل با مهاجرت ماشین‌های مجازی.

این دو معیار به صورت زیر تعریف می‌شوند:

$$SLATAH = \frac{۱}{N} \sum_{i=۱}^{N} \frac{T_{si}}{T_{ai}} \qquad (۸)$$

$$PDM = \frac{۱}{M} \sum_{j=۱}^{M} \frac{C_{dj}}{C_{rj}} \qquad (۹)$$

N تعداد منابع محاسباتی، T_{si} زمان کل در زمانی که ماشین فیزیکی i ام بهره‌ی ۱۰۰٪ را متحمل می‌شود که نقض کیفیت سرویس را ایجاد می‌کند. T_{ai} زمان کل ماشین فیزیکی i ام که در حالت فعال است، M تعداد ماشین‌های مجازی، C_{dj} نقض کارایی ماشین مجازی j ام که با مهاجرت ایجاد می‌شود را برآورد می‌کند. C_{rj} کل ظرفیت پردازنده درخواست شده توسط ماشین مجازی j ام در زمان اجرای آن است. در آزمایشات C_{dj} با ۱۰٪ از بهره پردازنده در MIPS در طول همه مهاجرت‌های ماشین مجازی j ام برآورد شده است. از آنجا که هر دو معیار SLATAH و PDM به طور مستقل و با اهمیت برابر سطح توافق‌نامه سطح سرویس نقض شده را تعیین می‌کنند، بنابراین یک معیار ترکیبی که شامل هر دو نقض کارایی با توجه به ماشین فیزیکی پربار و با توجه به مهاجرت ماشین مجازی است در نظر گرفته می‌شود [۱۹].

$$SLAV = SLATAH * PDM \qquad (۱۰)$$

[۲۵] SLA violation Time per Active Host

[۲۶] Performance Degradation due to Migrations

برای رویکرد پیشنهادی شبه کد زیر در نظر گرفته شده است.

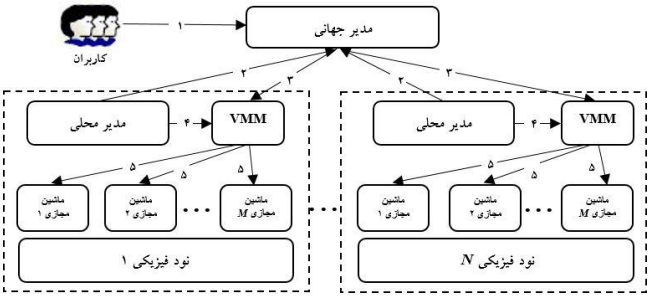
```
Pseudo-code Alghorithm
Input: hostList, vmList   Output: allocation of VMs
١) For each VM in VMlist do
٢)    allocatedHost ⟵ NULL;
٣)    MinFitness ⟵ MAX;
٤)  For each host in HostList do
٥)        if host has enough resource for vm then
٦)  fitness ⟵
(RequestedCPU(VM)/AllocatedMipsForVm(PM))
+(RequestedRAM(VM)/
AllocatedRAMForVm(PM))+
host.getMaxUtilization();
٧)            if fitness < MinFitness then
٨)                MinFitness ⟵fitness;
٩)                AllocatedHost ⟵ host;
١٠)            end if
١١)        end if
١٢)  if allocatedHost ≠ NULL then
١٣)        allocate vm to allocatedHost
١٤)  end if
١٥)  end for
١٦) return allocation;
١٧) end for
```

شکل ۳. شبه کد الگوریتم پیشنهادی

۴. تجزیه و تحلیل

۴. ۱ معماری سیستم

در این مقاله، معماری سیستم هدف یک محیط زیرساخت به‌عنوان سرویس می‌باشد که توسط یک دیتاسنتر در مقیاس بزرگ که شامل N نود فیزیکی ناهمگن می‌باشد ارائه شده است. مشخصات هر نود شامل پردازنده، حافظه و پهنای باند می‌باشد. سرورها دیسک‌های داخلی ندارند. حافظه ذخیره‌سازی به‌صورت ذخیره‌ساز تحت شبکه[۲۷] ارائه شده است که برای مهاجرت ماشین‌های مجازی به‌صورت آنلاین استفاده می‌شود. کاربران مستقل از هم درخواست‌هایی را ارسال می‌کنند که برای اجرای این درخواست‌ها نیاز به M ماشین مجازی ناهمگن با مشخصات مورد نیاز از پردازنده، حافظه و پهنای باند می‌باشد. کاربران به صورت مستقل از هم جریان‌های کاری متنوعی ایجاد می‌کنند. این جریان کاری شامل انواع مختلفی از برنامه‌ها مانند رایانش سریع[۲۸] و وب اپلیکیشن‌ها که از منابع به‌طور همزمان استفاده می‌کنند می‌باشد [۲۰].



شکل ۴: معماری سیستم [۲۰]

لایه نرم افزاری سیستم شامل مدیران محلی و جهانی می‌باشد همان‌طور که در شکل ۴ نشان داده شده است. مدیران محلی درون هر نود به‌عنوان یک ماژول VMM قرار داده شده‌اند. هدف آنها مانیتور کردن مداوم استفاده از پردازنده توسط هر نود، تغییر اندازه ماشین‌های مجازی بر اساس نیاز منابع از آنها و تصمیم‌گیری در مورد اینکه چه زمانی و کدام ماشین‌مجازی باید از نود مهاجرت داده شود (۴). مدیر جهانی داخل نود اصلی قرار دارد و اطلاعات را از مدیران محلی برای برقرار نگهداشتن استفاده از منابع جمع‌آوری می‌کند (۲). مدیر محلی دستوراتی را برای بهینه‌سازی فرآیند تخصیص صادر می‌کند (۳). ماژول‌های VMM تغییر اندازه نهایی و مهاجرت ماشین‌های مجازی و همچنین تغییرات در میزان مصرف انرژی در هر نود را انجام می‌دهند (۵) [۲۰].

۴. ۲ معماری پردازنده

در این مقاله، معماری پردازنده‌های سرورهای فیزیکی چند هسته‌ای می‌باشند. در این معماری یک پردازنده چند هسته‌ای با تعداد n هسته که هر کدام m میلیون دستورالعمل را در ثانیه می‌توانند اجرا کنند به‌عنوان یک پردازنده تک هسته‌ای با ظرفیت کل اجرای nm میلیون دستورالعمل در ثانیه می‌باشد. این کار به‌صورت نرم افزاری انجام می‌شود و همچنین ماشین‌های مجازی به هسته‌های پردازشی وابسته نیستند و می‌توانند با استفاده از الگوریتم زمانبندی اشتراک زمانی با استفاده از یک هسته دلخواه اجرا شوند. تنها محدودیت این روش این است که ظرفیت پردازنده مورد نیاز برای یک ماشین مجازی باید کمتر و یا برابر با ظرفیت یک هسته باشد. زمانی که ظرفیت پردازنده مورد نیاز برای یک ماشین مجازی بیشتر از ظرفیت یک هسته باشد نیاز به موازی اجرا شدن آن می‌باشد. به دلیل اینکه اطلاعاتی از قبل در مورد برنامه‌های در حال اجرا روی ماشین‌های مجازی نداریم و همچنین موازی سازی خودکار یک مشکل پیچیده تحقیقاتی می‌باشد ما موازی اجرا شدن ماشین‌های مجازی را در نظر نمی‌گیریم [۲۰].

[۲۷] Network Attached Storage (NAS)

[۲۸] High Performance Computing (HPC)

۴. ۳ شبیه‌سازی

ارزیابی الگوریتم پیشنهادی در زیرساخت مراکز داده ابری در مقیاس بزرگ و بر روی یک زیرساخت واقعی بسیار دشوار می‌باشد. در نتیجه برای اطمینان از تکرار آزمایش‌ها، شبیه‌سازها به عنوان راهی برای ارزیابی عملکرد پیشنهادی انتخاب شده‌اند. ابزار کلودسیم[۲۹] به‌عنوان یک بستر شبیه‌سازی انتخاب شده است [۲۱]. زیرا یک چارچوب شبیه‌سازی مدرن می‌باشد که برای محاسبات ابری طراحی شده است. این ابزار برخلاف سایر ابزارهای شبیه‌سازی، امکان مدل سازی محیط‌های مجازی، پشتیبانی از تأمین منابع درخواستی و مدیریت آنها را فراهم می‌کند. این شبیه‌ساز برای برنامه‌های آگاه از انرژی[۳۰] نیز توسعه یافته است. جدا از مدل سازی مصرف انرژی، قابلیت شبیه‌سازی برنامه‌های کاربردی با حجم کاری پویا نیز گنجانده شده است. برای ارزیابی روش پیشنهادی از حجم کاری دنیای واقعی برای ارزیابی بهتر، استفاده کرده‌ایم. حجم کاری استفاده شده، حجم کاری واقعی می‌باشد که توسط داده‌های Planet Lab ایجاد شده است. در نتیجه در ابزار کلودسیم، ما یک مرکز داده شبیه به ۸۰۰ گره فیزیکی ناهمگن را شبیه‌سازی کرده‌ایم که نیمی از آنها سرورهای HP ProLiant ML۱۱۰ G۴ می‌باشند و نیمی دیگر سرورهای HP ProLiant ML۱۱۰ G۵ می‌باشند. مشخصات این سرورها بر اساس [۲۲] در جدول ۱ آورده شده است. هر دسته از سرورها دارای سرعت پردازش، میزان حافظه، تعداد هسته و پهنای باند متفاوتی می‌باشند.

جدول ۱. مشخصات میزبان‌های فیزیکی در شبیه‌سازی

ماشین مجازی	Extra-large instance	Small instance	Micro instance
سرعت پردازش (MIPS)	۲۰۰۰	۱۰۰۰	۵۰۰
حافظه (MB)	۳۷۵۰	۱۷۰۰	۶۱۳
پهنای باند (Gbit / s)	۱	۱	۱
سایز (GB)	۲,۵	۲,۵	۲,۵

مشخصات انواع ماشین‌های مجازی استفاده شده بر اساس نمونه‌های واقعی آمازون EC۲ می‌باشد [۲۳]. با این تفاوت که همه ماشین‌های مجازی تک هسته‌ای هستند زیرا داده‌های حجم کاری مورد استفاده برای شبیه‌سازی از ماشین‌های مجازی تک هسته‌ای استفاده می‌کنند. در ابتدا، ماشین‌های مجازی با توجه به منابع مورد نیاز تخصیص داده می‌شوند. با این حال، در طول عمر شبیه‌سازی ماشین‌های مجازی با توجه به حجم کار از منابع کمتری استفاده می‌کنند و فرصت‌هایی برای تجمیع ماشین‌های مجازی ایجاد می‌شود. مشخصات انواع ماشین‌های مجازی استفاده شده در جدول ۲ آورده شده است. ماشین‌های مجازی از نظر سرعت پردازش و میزان حافظه متفاوت می‌باشند. همانطور که در جدول ۲ نشان داده شده است.

جدول ۲. مشخصات ماشین‌های مجازی در شبیه‌سازی

میزبان فیزیکی	سرعت پردازش (MIPS)	حافظه (MB)	تعداد هسته	پهنای باند (Gbit / s)	سایز (GB)
HP ProLiant ML۱۱۰ G۴	۱۸۶۰	۴۰۹۶	۲	۱	۱
HP ProLiant ML۱۱۰ G۵	۲۶۶۰	۴۰۹۶	۲	۱	۱

حجم کاری استفاده شده، حجم کاری واقعی می‌باشد که توسط داده‌های Planet Lab در ده روز مختلف دریافت شده است [۲۴]. این داده‌ها از داده‌های استفاده از پردازنده توسط بیش از ۱۰۰۰ ماشین مجازی در یک زمان از سرورهای واقع در بیش از ۵۰۰ مکان آورده شده است. برای این منظور، به طور تصادفی ۱۰ روز از بین جریان کار جمع‌آوری شده در ماه‌های مارس و آوریل ۲۰۱۱ انتخاب شده است. مشخصات داده‌ها در جدول ۳ نمایش داده شده است.

جدول ۳. مشخصات داده‌های جریان کار

تاریخ	تعداد ماشین‌های مجازی	Median	St. Dev
۲۰۱۱/۰۳/۰۳	۱۰۵۲	۶٪	۱۷٪/۰۹
۲۰۱۱/۰۳/۰۶	۸۹۸	٪۵	٪۱۶/۸۳
۲۰۱۱/۰۳/۰۹	۱۰۶۱	٪۴	٪۱۵/۵۷
۲۰۱۱/۰۳/۲۲	۱۵۱۶	٪۵	٪۱۲/۷۸
۲۰۱۱/۰۳/۲۵	۱۰۷۸	٪۶	٪۱۴/۱۴
۲۰۱۱/۰۴/۰۳	۱۴۶۳	٪۶	٪۱۶/۵۵
۲۰۱۱/۰۴/۰۹	۱۳۵۸	٪۶	٪۱۵/۰۹
۲۰۱۱/۰۴/۱۱	۱۲۳۳	٪۶	٪۱۵/۰۷
۲۰۱۱/۰۴/۱۲	۱۰۵۴	٪۶	٪۱۵/۱۵
۲۰۱۱/۰۴/۲۰	۱۰۳۳	٪۴	٪۱۵/۲۱

[۲۹] CloudSim

[۳۰] Energy-Aware

الگوریتم پیشنهادی و روش مورد مقایسه با استفاده از محیط توسعه نت‌بینز در نرم‌افزار کلودسیم ورژن ۳ کد نویسی شده‌اند و در سیستم عامل ویندوز ۶۴ بیتی و با حافظه ۸G اجرا گردیده است.

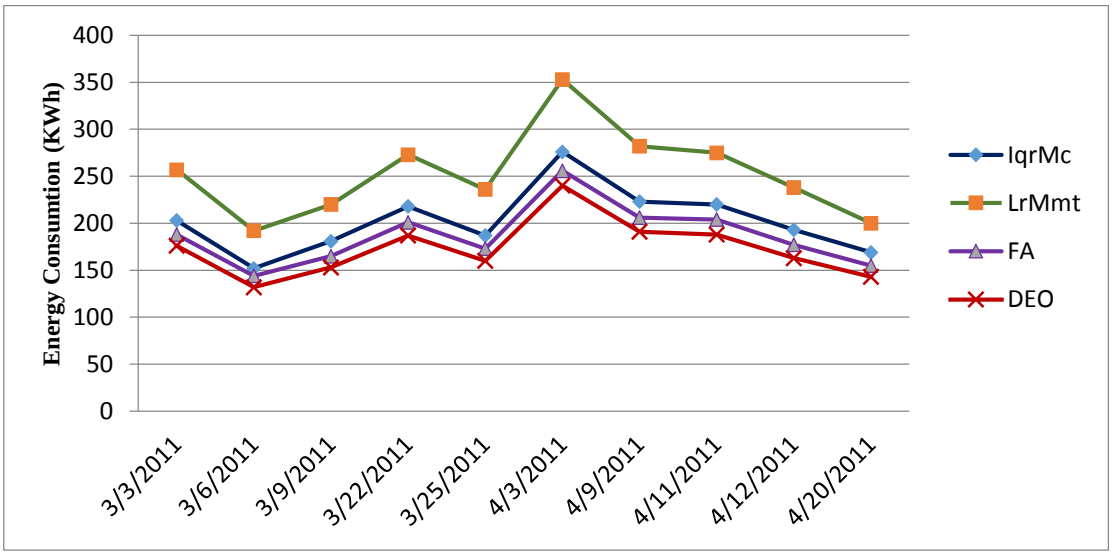
۴. ۴ نتایج شبیه‌سازی

معیارها و پارامترهای کیفیت سرویسی که در روش پیشنهادی در نظر گرفته شده است، انرژی مصرفی، تعداد مهاجرت‌ها، نقض توافق‌نامه سطح سرویس و زمان اجرای کل کارها می‌باشند. کارهایی که جهت مقایسه در نظر گرفته شده است روش IqrMc و روش LrMmt در مقاله [۲۰] و روش FA در مقاله [۱۴] می‌باشد. علت انتخاب این روش‌ها سازگاری بیشتر روش‌های مورد مقایسه با محیط شبیه‌سازی و استفاده از الگوریتم‌های فرا اکتشافی و غیر فرا اکتشافی می‌باشد. ایده و روش پیشنهاد شده در این مقاله، رویکردی جدید مبتنی بر الگوریتم تکاملی تفاضلی برای حل مساله تخصیص در فضای ابری می‌باشد که در رویکرد ارائه شده سعی می‌شود بر اساس تابع سودمندی چندهدفه تعریف شده و در نظر گرفتن بردارهای جهش و تقاطع بر اساس بهره‌وری از منابع و انرژی مصرفی، قصد داریم با استفاده بهینه از منابع و بهبود بهره‌وری در کاهش زمان اجرای کل کارها، انرژی مصرفی، تعداد مهاجرت‌ها و نقض کیفیت سرویس تأثیرگذار باشیم. برای بررسی روش پیشنهادی و روش‌های مورد مقایسه از شبیه‌ساز کلودسیم استفاده شده است.

در ادامه نمودارهای حاصل از شبیه‌سازی بر روی داده‌های Planet Lab برای پارامترهای کیفیت سرویس مورد نظر آورده شده است.

۴. ۵. ۱ بررسی انرژی مصرفی

معیار انرژی مصرفی میزان اتلاف انرژی کل مراکز داده که شامل مجموع انرژی میزبان‌های فیزیکی می‌باشد را نمایش می‌دهد. این اتلاف انرژی، ناشی از اجرای وظایف مورد نظر بر روی هسته‌های پردازشی مختلف است. شکل ۵، انرژی مصرفی روش پیشنهادی DEO و روش‌های مورد مقایسه را بر روی داده‌های Planet Lab نمایش می‌دهد

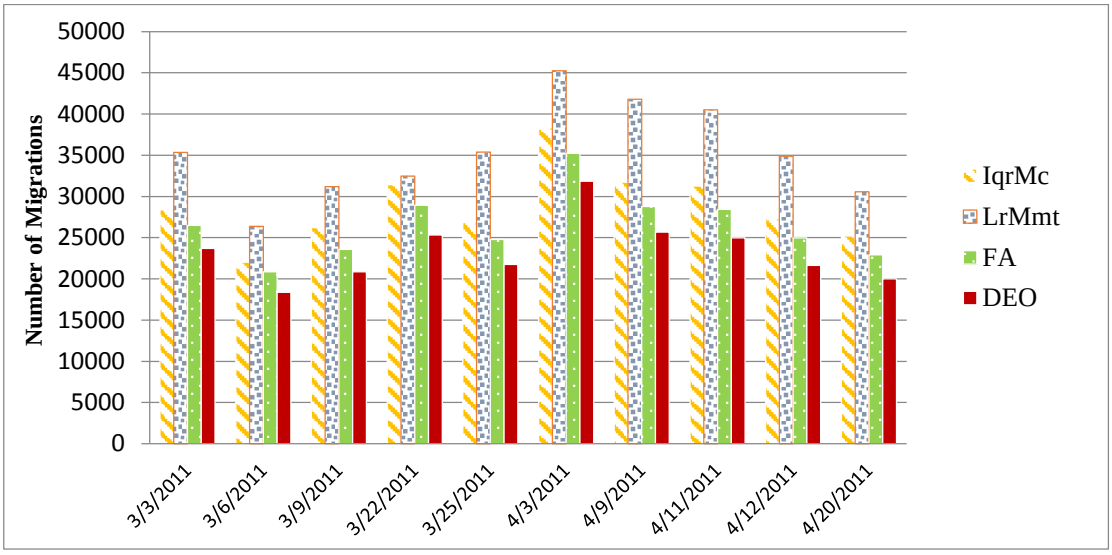


شکل ۵. انرژی مصرفی

با توجه به شکل ۵، روش پیشنهادی کمترین میزان مصرف انرژی در بین سایر الگوریتم‌ها در تمام ده روز را دارا می‌باشد. داده‌ها در زمان‌های متفاوتی بر روی محور افقی و انرژی مصرفی برحسب کیلو وات بر روی محور عمودی نشان داده شده است. با توجه به نمودار فوق مشخص است که، عملکرد روش پیشنهادی DEO در کاهش انرژی مصرفی در مقایسه با روش‌های مورد مقایسه IqrMc، LrMmt، FA بهتر می‌باشد. همانطور که در نمودار بالا نمایش داده می‌شود در روش پیشنهادی سعی شده است با استفاده از الگوریتم تکاملی تفاضلی تخصیص مؤثر و مناسبی از منابع محاسباتی انجام شود. از آنجایی که پردازنده بیشترین تأثیر را در مصرف انرژی دارد در رویکرد ارائه شده که بر اساس تابع سودمندی چندهدفه تعریف شده است با در نظر گرفتن پردازنده در تابع هدف پیشنهادی باعث می‌شود میزبانی که پردازنده آزاد بیشتری دارد جهت تخصیص انتخاب شود در واقع این میزبان احتمال پربار شدن آن کمتر است و از مهاجرت اضافی که در افزایش انرژی مصرفی تأثیرگذار است جلوگیری می‌کند و همچنین از تعداد سرورهای فعال کمتری استفاده خواهد شد. در نتیجه ما توانسته‌ایم مصرف انرژی را در مقایسه با مقالات مورد مطالعه کاهش دهیم. انرژی مصرفی در روش پیشنهادی DEO به میزان ۱۷٪ نسبت به روش IqrMc، ۴۶٪ نسبت به روش LrMmt و ۸٪ نسبت به روش FA بهبود داشته است.

۴. ۵. ۲ بررسی تعداد مهاجرت‌ها

یکی از مهمترین عواملی که بر مصرف انرژی و نقض توافق‌نامه سطح سرویس تأثیر می‌گذارد تعداد مهاجرت‌ها می‌باشد. شکل ۶ مقایسه بین روش DEO با IqrMc، LrMmt وFA از نظر تعداد مهاجرت‌ها را نمایش می‌دهد.

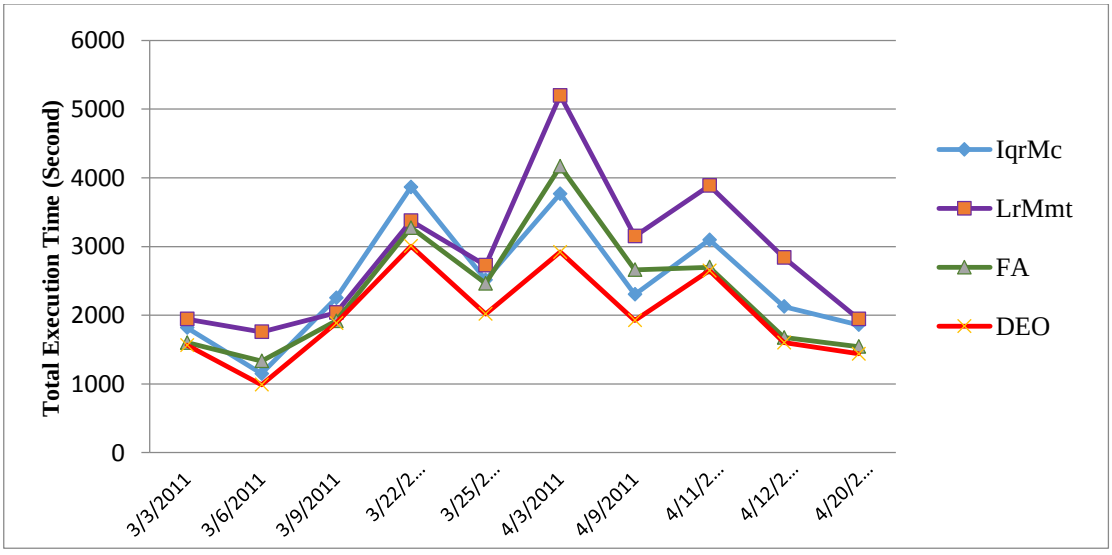


شکل ۶. تعداد مهاجرت‌ها

در نمودار بالا مشاهده می‌شود که روش DEO حداقل تعداد مهاجرت را در تمام ده روز دارد. از آنجایی که میزان حافظه بر روی مهاجرت تأثیر دارد در رویکرد ارائه شده که بر اساس تابع سودمندی چندهدفه تعریف شده است با در نظر گرفتن حافظه در تابع هدف پیشنهادی باعث می‌شود میزبانی که حافظه آزاد بیشتری دارد جهت تخصیص انتخاب شود در واقع این میزبان احتمال پر بار شدن آن کمتر است و از مهاجرت‌های اضافی جلوگیری می‌کند و در انتخاب مقصد مناسب برای تجمیع با استفاده از تابع هدف چند معیاره در الگوریتم پیشنهادی از مهاجرت‌های اضافی جلوگیری می‌کند. در نتیجه در روش پیشنهادی ما توانسته‌ایم تعداد مهاجرت‌ها را در مقایسه با مقالات مورد مقایسه کاهش دهیم. شکل ۶ تعداد مهاجرت‌ها را در این ده روز نمایش می‌دهد. رویکرد ارائه شده بر اساس تابع سودمندی چندهدفه تعریف شده توانسته است در کاهش تعداد مهاجرت‌ها در مقایسه با مقالات مورد مطالعه نسبت به روش IqrMc به میزان ۲۳٪، نسبت به روش LrMmt به میزان ۵۱٪ و نسبت به روش FA به میزان ۱۳٪ بهبود داشته باشد.

۴. ۵. ۳ بررسی زمان اجرای کل

یکی دیگر از اهداف الگوریتم پیشنهادی، به حداقل رساندن زمان اجرای کل می‌باشد. شکل ۷، زمان اجرای کل روش پیشنهادی و روش مقالات مورد مقایسه را بر روی داده‌های Planet Lab نمایش می‌دهد.

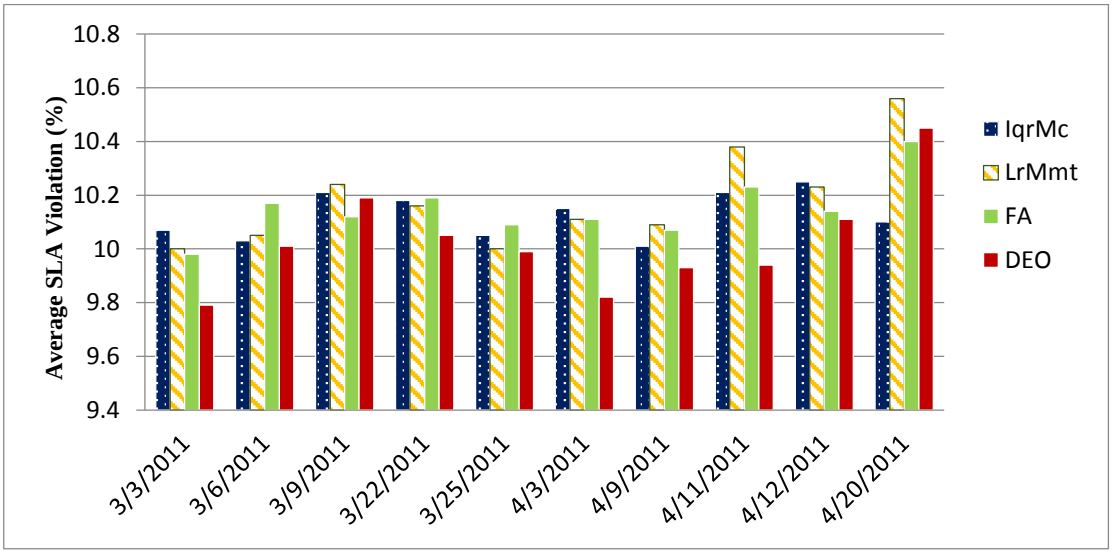


شکل ۷. زمان اجرای کل

با توجه به شکل ۷، روش پیشنهادی کمترین میزان زمان اجرای کل در بین سایر الگوریتم‌ها در بین روزهای مختلف را دارا می‌باشد. داده‌ها در روزهای متفاوت بر روی محور افقی و زمان اجرا برحسب ثانیه بر روی محور عمودی نشان داده شده است. با توجه به نمودار فوق مشخص می‌باشد که، عملکرد روش پیشنهادی DEO در کاهش زمان اجرای کل در مقایسه با روش‌های مورد مقایسه IqrMc، LrMmt و FA بهتر می‌باشد. در رویکرد ارائه شده که بر اساس تابع سودمندی چندهدفه تعریف شده است با در نظر گرفتن پردازنده و حافظه در تابع هدف پیشنهادی باعث می‌شود میزبانی که پردازنده آزاد و حافظه بیشتری دارد جهت تخصیص انتخاب شود در واقع این میزبان احتمال پربار شدن آن کمتر است و از مهاجرت اضافی که در افزایش زمان اجرای کل تأثیرگذار است جلوگیری می‌کند و از تعداد سرورهای فعال کمتری استفاده خواهد شد که باعث بهینه‌سازی فرآیند تخصیص می‌شود. در نتیجه زمان اجرای کل در روش پیشنهادی DEO نسبت به روش IqrMc به میزان ۲۳٪، نسبت به روش LrMmt به میزان ۴۶٪ **و** نسبت به روش FA به میزان ۱۸٪ بهبود داشته است.

۴. ۵. ۴ بررسی نقض توافق‌نامه سطح سرویس

یکی دیگر از معیارهای مهم که در بهبود بهره‌وری سیستم‌های ابری نقش حیاتی دارد، معیار کیفیت سرویس می‌باشد که در شرکت‌های ارائه دهنده ابر نقش بسیار مهمی دارد. هر چه بتوانیم در شناسایی سرور پر بار در تجمیع روش مناسب‌تری را ارائه دهیم بهتر می‌توانیم در کاهش نقض توافق‌نامه سطح سرویس که به معیارهای تعداد مهاجرت‌ها و پربار بودن سرورها بستگی دارد، تأثیرگذار باشیم. شکل ۸ نتایج ارزیابی میانگین نقض توافق‌نامه سطح سرویس روش پیشنهادی و سایر روش‌ها را در روزهای مختلف بر روی داده‌های Planet Lab نمایش می‌دهد.



شکل ۸. میانگین نقض توافق‌نامه سطح سرویس

با توجه به شکل ۸، روش پیشنهادی DEO در تاریخ‌های ۳/۹/۲۰۱۱ و ۴/۲۰/۲۰۱۱ نتوانسته عملکرد بهتری را نسبت به سایر الگوریتم‌ها نشان دهد. میزان درصد میانگین نقض توافق‌نامه سطح سرویس بر روی محور عمودی نمایش داده شده است. با توجه به نمودار فوق مشخص می‌باشد که عملکرد روش DEO در کاهش نقض توافق‌نامه سطح سرویس در اکثر روزها در مقایسه با روش‌های مورد مقایسه بهتر می‌باشد. دلیل این کاهش تخصیص مناسب و جلوگیری از پرباری و کاهش مهاجرت‌های اضافی با تعریف تابع هدف چند معیاره پیشنهادی می‌باشد. در نتیجه در روش پیشنهادی ما توانسته‌ایم نسبت به روش IqrMc به میزان ۱٪، نسبت به روش LrMmt به میزان ۲٪ و نسبت به روش FA به میزان ۱٪ در کاهش نقض توافق‌نامه سطح سرویس تأثیرگذار باشیم.

۵. نتیجه‌گیری و پیشنهادات

برای به حداکثر رساندن مزایای ارائه دهندگان خدمات ابری، لازم است مصرف انرژی بدون هیچ گونه عارضه جانبی یا تضعیف توافق‌نامه سطح سرویس در مراکز ابری کاهش پیدا کند. الگوریتم‌های متنوعی برای تخصیص وظایف به منابع در محیط ابر پیشنهاد شده است که اغلب آن‌ها معیارهایی همچون توزیع بار متعادل، تخصیص بهینه‌ی منابع، کاهش زمان اجرا و کاهش انرژی مصرفی را در نظر نمی‌گیرند. برای بهبود انرژی مصرفی علاوه بر فرایند تخصیص، مدیریت منابع با استفاده از تجمیع ماشین‌های مجازی نقش مؤثری را در محاسبات ابری ایفا می‌کند. به همین دلیل به منظور زمانبندی و تخصیص بهینه وظایف به منابع در محیط ابر یک رویکرد مبتنی بر الگوریتم تکاملی تفاضلی چندهدفه برای مساله زمانبندی کارها، حل مساله تخصیص و بهبود بهره‌وری از منابع در محیط رایانش ابری ارائه می‌شود. در روش پیشنهادی سعی می‌شود که بر اساس تابع

سودمندی چندهدفه تعریف شده و در نظر گرفتن بردارهای جهش و تقاطع بر اساس بهره‌وری از منابع و انرژی مصرفی قصد داریم در کاهش اهدافی چون زمان اجرای کل و مصرف انرژی تأثیرگذار باشیم. ما الگوریتم پیشنهادی را از طریق شبیه‌سازی‌های گسترده در شبیه‌ساز کلودسیم با آزمایش بر روی حجم کار بیش از هزار ماشین مجازی بر روی داده‌های Planet Lab ارزیابی کرده‌ایم. نتایج شبیه‌سازی نشان دهنده این می‌باشد که الگوریتم پیشنهادی توانسته است معیارهای مصرف انرژی را نسبت به سه الگوریتم مورد مقایسه به طور میانگین به میزان ۲۳ درصد، تعداد مهاجرت‌ها را به طور میانگین به میزان ۲۹ درصد، زمان اجرای کل را به طور میانگین به میزان ۲۹ درصد، نقص توافق‌نامه سطح سرویس را به طور میانگین به میزان ۱ درصد نسبت به روش‌های مورد مقایسه بهبود دهد. در این صورت استفاده از رویکرد پیشنهادی در مراکز ابری منجر به سرویس‌های بهتر و مناسب به مشتریان این مراکز خواهد شد. لذا برای کارهای آینده در راستای این پژوهش در نظر داریم موارد زیر را مورد مطالعه قرار دهیم:

- استفاده از چند الگوریتم فرا ابتکاری به صورت ترکیبی با استفاده از تابع هدف ارائه شده در روش پیشنهادی.
- ترکیب پردازش ابری با پردازش مه به منظور جلوگیری از تأخیر و استفاده از تابع هدف ارائه شده در روش پیشنهادی به منظور تخصیص کارآمدتر.
- در نظر گرفتن معیارهای نقض کیفیت سرویس و چارچوب‌های امنیتی در روش پیشنهادی.
- بررسی روش پیشنهادی در شبکه‌های خودرویی به منظور بهبود انرژی و کاهش هزینه.
- ترکیب روش پیشنهادی با شبکه‌های مه و اینترنت اشیا به منظور بهبود بهره‌وری و کاهش هزینه‌ها در این شبکه‌ها.

مراجع

[۱] H. Yuan, J. Bi, and M. Zhou, "Spatial task scheduling for cost minimization in distributed green cloud data centers," *IEEE Transactions on Automation Science and Engineering*, vol. ۱٦, no. ۲, pp. ۷۲۹-۷٤۰, ۲۰۱۸.

[۲] A. Arunarani, D. Manjula, and V. Sugumaran, "Task scheduling techniques in cloud computing: A literature survey," *Future Generation Computer Systems*, vol. ۹۱, pp. ٤۰۷-٤۱٥, ۲۰۱۹.

[۳] Y. Li, S. Wang, X. Hong, and Y. Li, "Multi-objective task scheduling optimization in cloud computing based on genetic algorithm and differential evolution algorithm," in ۲۰۱۸ ۳۷*th Chinese Control Conference (CCC)*, ۲۰۱۸: IEEE, pp. ٤٤۸۹-٤٤۹٤.

[٤] E. H. Houssein, A. G. Gad, Y. M. Wazery, and P. N. Suganthan, "Task scheduling in cloud computing based on meta-heuristics: Review, taxonomy, open challenges, and future trends," *Swarm and Evolutionary Computation*, vol. ٦۲, p. ۱۰۰۸٤۱, ۲۰۲۱.

[٥] E. Arianyan, H. Taheri, and V. Khoshdel, "Novel fuzzy multi objective DVFS-aware consolidation heuristics for energy and SLA efficient resource management in cloud data centers," *Journal of Network and Computer Applications*, vol. ۷۸, pp. ٤۳-٦۱, ۲۰۱۷.

[٦] D. Maharana, B. Sahoo, and S. Sethi, "Energy-efficient real-time tasks scheduling in cloud data centers," *International Journal of Science Engineering and Advance Technology, IJSEAT*, vol. ٤, no. ۱۲, pp. ۷٦۸-۷۷۳, ۲۰۱۷.

[۷] H. Wang and H. Tianfield, "Energy-aware dynamic virtual machine consolidation for cloud datacenters," *IEEE Access*, vol. ٦, pp. ۱٥۲٥۹-۱٥۲۷۳, ۲۰۱۸.

[۸] Y. Li, S. Wang, X. Hong, and Y. Li, "Multi-objective task scheduling optimization in cloud computing based on genetic algorithm and differential evolution algorithm." pp. ٤٤۸۹-٤٤۹٤.

[۹] S. Srichandan, T. A. Kumar, and S. Bibhudatta, "Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm," *Future Computing and Informatics Journal*, vol. ۳, no. ۲, pp. ۲۱۰-۲۳۰, ۲۰۱۸.

[۱۰] J. Li, J. Liu, and J. Wang, "An improved differential evolution task scheduling algorithm based on cloud computing," in ۲۰۱۸ ۱۷*th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, ۲۰۱۸: IEEE, pp. ۳۰-۳٥.

[۱۱] X. Zhang *et al.*, "Energy-aware virtual machine allocation for cloud with resource reservation," *Journal of Systems and Software*, vol. ۱٤۷, pp. ۱٤۷-۱٦۱, ۲۰۱۹.

[۱۲] A. Rehman, S. S. Hussain, Z. ur Rehman, S. Zia, and S. Shamshirband, "Multi- objective approach of energy efficient workflow scheduling in cloud environments,"

Concurrency and Computation: Practice and Experience, vol. ۳۱, no. ۸, p. e٤٩٤٩, ۲۰۱۹.

[۱۳] K. Naik, G. Meera Gandhi, and S. Patil, "Multiobjective virtual machine selection for task scheduling in cloud computing," in *Computational Intelligence: theories, applications and future directions-volume I*: Springer, ۲۰۱۹, pp. ۳۱۹-۳۳۱.

[۱٤] E. Aloboud and H. Kurdi, "Cuckoo-inspired job scheduling algorithm for cloud computing," *Procedia Computer Science*, vol. ۱٥۱, pp. ۱۰۷۸-۱۰۸۳, ۲۰۱۹.

[۱٥] M. Aruna, D. Bhanu, and S. Karthik, "An improved load balanced metaheuristic scheduling in cloud," *Cluster Computing*, vol. ۲۲, no. ٥, pp. ۱۰۸۷۳-۱۰۸۸۱, ۲۰۱۹.

[۱٦] N. Mc Donnell, E. Howley, and J. Duggan, "Dynamic virtual machine consolidation using a multi-agent system to optimise energy efficiency in cloud computing," *Future Generation Computer Systems*, vol. ۱۰۸, pp. ۲۸۸-۳۰۱, ۲۰۲۰.

[۱۷] N. Khattar, J. Singh, and J. Sidhu, "An energy efficient and adaptive threshold VM consolidation framework for cloud environment," *Wireless Personal Communications*, vol. ۱۱۳, no. ۱, pp. ۳٤۹-۳٦۷, ۲۰۲۰.

[۱۸] R. Shaw, E. Howley, and E. Barrett, "Applying reinforcement learning towards automating energy efficient virtual machine consolidation in cloud data centers," *Information Systems*, vol. ۱۰۷, p. ۱۰۱۷۲۲, ۲۰۲۲.

[۱۹] S. Mustafa, K. Bilal, S. U. R. Malik, and S. A. Madani, "SLA-aware energy efficient resource management for cloud environments," *IEEE Access*, vol. ٦, pp. ۱٥۰۰٤-۱٥۰۲۰, ۲۰۱۸.

[۲۰] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency and Computation: Practice and Experience*, vol. ۲٤, no. ۱۳, pp. ۱۳۹۷-۱٤۲۰, ۲۰۱۲.

[۲۱] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and experience*, vol. ٤۱, no. ۱, pp. ۲۳-٥۰, ۲۰۱۱.

[۲۲] Standard Performance Evaluation Corporation. https://www.spec.org/power_ssj۲۰۰۸/results/

[۲۳] Amazon EC۲ Instance Types. https://aws.amazon.com/ec۲/instance-types/

[۲٤] K. Park and V. S. Pai, "CoMon: a mostly-scalable monitoring system for PlanetLab," *ACM SIGOPS Operating Systems Review*, vol. ٤۰, no. ۱, pp. ٦٥-۷٤, ۲۰۰٦.