

ارزیابی عملکرد مسائل مجازی سازی مبتنی بر کانتینر در زمان بندی ریزسرویس ها در ابر

حسین علی ناقد کبه

کارشناسی ارشد کامپیوتر دانشگاه آزاد اسلامی واحد علوم تحقیقات تهران

چکیده

رایانش ابری محصول پیشرفت های مجازی سازی، اینترنت، سرورهای ارزان و ... در کنار یکدیگر می باشد که در سال های اخیر توسعه خیره کننده ای داشته است. در واقع رایانش ابری مدلی است برای دسترسی آسان بر اساس تقاضای کاربر از طریق شبکه به مجموعه ای از منابع محاسباتی قابل تغییر و تنظیم خواهد بود. این فناوری نسبتاً نوظهور با چالش های متعددی مواجه است که در این میان زمان بندی یکی از چالش های مهم بشمار می آید. الگوهای مختلفی برای زمان بندی منابع، پیشنهاد شده که هر یک از آن ها دارای محاسن و محدودیت های خاصی هستند؛ اما تاکنون الگوهای مبتنی بر کانتینر کمتر مورد توجه پژوهشگران قرار گرفته است. کانتینرها فناوری مجازی سازی سبکی هستند که امروزه برای ارائه ریزسرویس ها رایج شده است و نسبت به دیگر فناوری ها نه تنها کارها را برای کاربران آسان کرده است؛ بلکه برای شرکت ها نیز مزیت های فراوانی در پی دارد. با توجه به قابلیت های تکنیک موصوف در این پایان نامه راه حلی مبتنی بر کانتینر و الگوریتم بهینه سازی عقاب طلایی برای زمان بندی ابر ارائه شده است. روش پیشنهادی در محیط نرم افزاری متلب پیاده سازی شده و کاری آن در شش شاخص زمان پاسخگویی، زمان اتمام وظایف، انرژی مصرفی، تخطی از زمان تعیین شده، تعداد برنامه های اجرا شده در خارج از زمان تعیین شده، سر بار محاسباتی (زمان زمان بندی) مورد بررسی قرار گرفت. بررسی اجرای آزمایش های مختلف نشان داد که مدل پیشنهادی در تمامی شاخص ها نسبت به طرح پایه که یک رویکرد مبتنی بر استراتژی حریمانه و آگاه از اولویت می باشد؛ عملکرد بهتری دارد. به نحوی که در شاخص زمان پاسخگویی ۸ درصد، در شاخص زمان اتمام کارها ۷ درصد، در شاخص انرژی مصرفی ۲۷ درصد، در شاخص تخطی از زمان تعیین شده ۲۴ درصد، در شاخص برنامه های اجرا شده در خارج از زمان ضرب الاجل ۲۲ درصد و نهایتاً در شاخص پیچیدگی محاسباتی ۷ درصد، عملکرد آن بهبود داشته است.

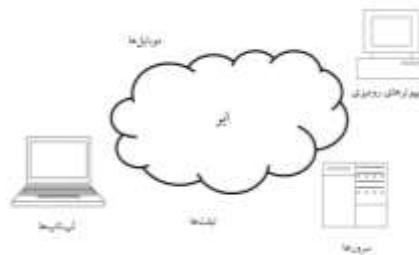
کلمات کلیدی: رایانش ابری، زمان بندی، کانتینر، الگوریتم بهینه سازی عقاب طلایی.

مقدمه

رایانش ابری به وسیله اینترنت، سرویس‌ها و ابزارهای مختلفی از جمله ذخیره‌سازی داده‌ها، سرورها، نرم‌افزار، شبکه، پایگاه‌های داده و... ارائه خدمات می‌دهد. به تعریفی ذخیره‌سازی در ابر، این فرصت را محیا می‌کند تا از راه دور در پایگاه‌های داده به جای استفاده از هارد دیسک‌های اختصاصی و... توانایی ذخیره فراهم شود.

که دیگر برای استفاده از این اطلاعات وابسته به مکانی محدود نباشد و با اتصال به اینترنت در هر زمان و مکانی توانایی دسترسی به اطلاعات محیا شود. ابر با سرویس‌های منعطف به خود، باعث تحولاتی بزرگی در دنیای فناوری اطلاعات شده است و با ظهور ایده‌های جدید و به اجرا پیوستن برخی از آن‌ها باعث این شده است تا این صنعت به طور همگانی‌تر در اختیار مصرف‌کننده‌ها و کاربران زیادی قرار بگیرد و کاربردی تر شود [۱].

امروزه خدمات ابری رو به گسترش چشمگیری است و جامعه‌ای از اکثریت عموم مردم جهان را شامل می‌شود که هرکس به طریقی و معیاری از افرادی که اطلاعات را به وجود می‌آورند تا مالکان وب سایت‌ها و کاربرانی که بسته به نیاز خود از این اطلاعات استفاده می‌کنند، همه به نوعی درگیر استفاده از خدمات هستند. ابر با ارائه راهکارهای جدید میزبانی از طریق اینترنت ظهور کرده و نظر خیلی از متخصصین و پژوهشگران این عرصه را به خود جلب کرده است [۲]. هرچند ابر یک فناوری تازه و نو است اما در مقاله‌ای جان مکاری اینگونه نوشته است که روزی ابر به عنوان عنصری پنجم از ان نام خواهند برد که در دسترس همه به صورت عمومی خواهد بود [۳].



شکل ۱- نمای کلی ابر

سرویس‌های ابری که در اختیار کاربران و شرکت‌ها و... قرار داده می‌شود این توانایی را ایجاد می‌کند تا برحسب نیاز خود، نرم‌افزارهای مربوطه را از طریق اینترنت دریافت کنند [۴].

اهداف تحقیق

۱. بررسی مسائل زمان‌بندی در مجازی‌سازی مبتنی بر کانتینر
۲. شناسایی و گروه‌بندی عوامل محرک و بازدارنده و موثر بر مسائل مجازی‌سازی مبتنی بر کانتینر
۳. ارائه راهکارهایی جهت افزایش سطح درک و آگاهی سازمان‌ها برای روی آوردن و پذیرش فناوری رایانش ابری مبتنی بر کانتینر
۴. بهینه‌سازی سرویس و کم کردن هزینه‌های شرکت‌های ارائه دهنده سرویس و افزایش استقبال کاربران از ابر و خدمات کانتینر
۵. کمک به فرآیند تصمیم‌گیری برای شرکت‌هایی که قصد اتخاذ و استفاده از محاسبات ابری را دارند
۶. به حداقل رساندن تعداد سیستم‌عامل‌های مختلف در یک سخت‌افزار

سوال‌های تحقیق

۱. عوامل موثر بر مسائل مجازی‌سازی مبتنی بر کانتینر کدام‌اند؟
۲. چه عواملی باعث استقبال مسائل مجازی‌سازی مبتنی بر کانتینر می‌شود؟

۳. هدف اصلی و کاربردی استفاده از کانتینر به جای مجازی‌سازی در سخت‌افزار چیست؟

فرضیه های تحقیق

۱. رشد و توسعه مسائل مجازی‌سازی مبتنی بر کانتینر باعث استفاده آسان‌تر و فراگیرتر شدن در استفاده از ابر می‌شود.
۲. نحوه‌ی اجرا برنامه و کدنویسی و پردازش صورت گرفته آسان‌تر و استفاده از نرم‌افزارهای مخصوص کاهش می‌یابد
۳. کاهش استفاده از سخت‌افزارها
۴. افزایش کیفیت خدمات‌رسانی
۵. در دسترس قرار گرفتن راحت کدهای نوشته شده بین کاربران و توسعه کدها از سمت کاربران

تحقیقات انجام شده

Park و همکاران در سال ۲۰۱۳، خدمات مبتنی بر مکان ارائه دادند بخاطر محدودیت‌های تلفن همراه که با تقسیم بندی وظایف، کارهای سبک درون تلفن همراه انجام شود و مابقی در ابر [۵]. Buchert و همکاران در سال ۲۰۱۵، پژوهشی بر مبنای روش زمان بندی آگاه به هزینه ارائه دادند، از آنجا که هدف اصلی ارائه دهندگان ابر دستیابی به ساختار پولی برای در اختیار گذاشتن منابع می باشد لذا روش زمان بندی که تعهدات اجرا را با هزینه پولی بهینه در نظر بگیرد ضروری می باشد (بین زمان اجرا و هزینه پولی پرداخت شده، تعادل برقرار می‌شود) [۶].

یوسف در سال ۲۰۱۳، معماری ترکیبی جهت انجام کارهای از جمله پردازش؛ داده کاوی و اجرای برنامه و... معرفی کردند که از لحاظ تنوع و دسترسی، خدمات یکسان تصور شود [۷]. بهره‌پور و رضایی زاده در سال ۱۳۹۵، آن‌ها تحقیقی در رابطه با الگوریتم کارآمد برای زمان بندی وظایف در محیط محاسبات ابری انجام دادند و نتایج کار آن‌ها نشان داد که تعادل بار پردازنده‌ها منجر به کاهش زمان تکمیل هر پردازنده و افزایش توان عملیاتی می‌شود. [۸]. Al Ali و همکاران در سال ۲۰۱۴، معماری ترکیبی از چهار لایه معرفی کردند که دولایه وظیفه پردازش و دولایه وظیفه جمع‌آوری اطلاعات [۹].

حسینی اهنگر و همکاران در سال ۱۳۹۹، تحقیقی در زمینه ارائه یک روش زمان بندی وظایف تحمل پذیر خطا برای استفاده بهینه از منابع در محیط محاسبات ابری انجام دادند و نتایج کار آن‌ها نشان داد که روش پیشنهادی توانسته است بهینه‌سازی بیشتری را در استفاده از منابع نسبت به تکرار به ارمغان بیاورد. [۱۰]. Hwang و همکاران در سال ۲۰۱۱، در محاسبات فراگیر یک روش زمان بندی دارای الگوریتم ساده ارائه دادند که درخواست‌ها زمانی که روی یک پردازنده اجرا شد، درخواست در صف ورودی به وسیله زمانبند فراخوانی شود [۱۱].

AlFailakawi و همکاران در سال ۲۰۲۰، پژوهشی با عنوان بررسی و ارزیابی تکنیک‌های زمان بندی کانتینر انجام دادند آن‌ها تکنیک‌ها را براساس نوع الگوریتم بهینه‌سازی به کار رفته برای تولید به چهار دسته تقسیم کردند. سپس برای هر کلاس از الگوریتم‌های زمان بندی، مزایا و مشکلات کلیدی را تجزیه و تحلیل و شناسایی کردند [۱۲]. Kovachev و همکاران در سال ۲۰۱۲، توانستند امکانی را به وجود بیاورند که برنامه‌های تلفن همراه در ابر جهت بارگذاری اجرا کند [۱۳]. Shiraz و همکاران در سال ۲۰۱۴، فرایندی را به وجود آوردند که براساس تقاضاهای آینده، از راه دور منابع بر روی تلفن همراه قابلیت بارگذاری داشته باشد [۱۴].

Prado و همکارانش در سال ۲۰۲۰، پژوهشی با عنوان بررسی چالش‌ها و فرصت‌های زمان بندی کانتینرهای هوشمند برای ارائه ریز سرویس‌ها در شبکه‌های اینترنت اشیا ابری مورد بررسی قرار گرفت و چالش اصلی در زمان بندی کانتینرهای Docker در شبکه‌های اینترنت اشیا مطرح و مورد بحث قرار گرفت. سپس، در ارتباط با ادغام زمان بندی کانتینر هوشمند مبتنی بر محاسبات نرم در پلت‌فرم‌های مدیریت کانتینر و زمان بندی کانتینرهای هوشمند خاص برای رابط‌های مختلف در شبکه‌های اینترنت اشیا، هدف این کار پشتیبانی از تخصیص بهینه ریز سرویس‌های ارائه شده توسط ارائه دهندگان خدمات

ابری اصلی امروزی و میلیون ها کاربر در سرتاسر جهان از آن استفاده می کنند [۱۵]. Roostaei و همکاران در سال ۲۰۱۶، براساس پهنای باند، میزان داده‌های ارسالی از سمت کاربر به کمک زنجیره مارکوف یک بهینه راه حل پیدا کند [۱۶]. Mohammadi و همکاران در سال ۲۰۱۶، برای ارسال برنامه‌های مناسب از بین برنامه‌های چند مولفه ای یک الگوریتم زمان‌بندی و بارگذاری ارائه دادند [۱۷].

Xavier و همکارانش در سال ۲۰۱۳، پژوهشی با عنوان ارزیابی عملکرد مجازی سازی مبتنی بر کانتینر برای محیط های محاسباتی با عملکرد بالا انجام دادند. آن ها تعدادی آزمایش را به منظور انجام ارزیابی عملکرد عمیق مجازی سازی مبتنی بر کانتینر برای HPC انجام دادند. همچنین مبادله بین عملکرد و ایزوله را در سیستم‌های مجازی سازی مبتنی بر کانتینر ارزیابی کردند و آن‌ها را با Xen مقایسه کردند، که نماینده‌ای از سیستم‌های مجازی سازی مبتنی بر هایپروایزر سنتی است که امروزه استفاده می شود [۱۸]. Tout و همکاران در سال ۲۰۱۶، در مقاله خود کیفیت چندین هدف از جمله میزان استفاده از پردازنده و زمان اجرا و... بررسی و ارتقا دادند [۱۹]. Fehling و همکاران در سال ۲۰۱۴، یک زمانبند جدید پیشنهاد دادند که نتایج حاصل از کارشان بیانگر این بود که زمانبند پیشنهادی آن‌ها از زمانبندهای دارای موجودی بهتر و منابع مولدتر، زمان تکمیل کمتر و متعادل سازی بار بهتر است [۲۰]. Yang و همکاران در سال ۲۰۱۶، در پژوهش خود بخش‌بندی زمان را در محیط‌های ابری تلفن همراه برای اجرای برنامه مورد بررسی قرار دادند؛ که نتایج بدست آمده حاکی از آن بود که براساس پهنای باند، به طور متناوب بخش‌بندی به روزرسانی شود [۲۱].

روش کار

مدل سیستم

امروزه داکر رایج ترین نوع کانتینرهای کاربردی است. در مقایسه با کانتینرها در سطح سیستم، به عنوان مثال، OpenVZ و LXC، کانتینرهای برنامه محور مانند Docker برای چارچوب های میکروسرویس مناسبتر هستند و به این ترتیب، در استقرار معماری های ابر-مه IoT-شتاب بیشتری به دست می آورند. کانتینرهای داکر یک سیستم فایل سازمان یافته در لایه‌ها را ادغام می کنند تا هسته سیستم عامل میزبان را به اشتراک بگذارند، چیزی که باعث عملکرد کارآمدتر در استقرار نمونه‌ها می شود. داکر بر اساس تصاویر، یا به طور خاص، بر روی عکس های فوری از یک سیستم عامل است. برای ایجاد یک تصویر جدید، باید از یک تصویر پایه شروع کرد، تغییرات مورد نظر را انجام داد و آن‌ها را ذخیره کرد. تصاویر تولید شده را می توان در سوابق عمومی یا خصوصی برای استفاده توسعه دهندگان دیگر به اشتراک گذاشت که فقط باید آن تصاویر را استخراج کنند. استفاده از تصاویر برای ایجاد عکس های فوری از سیستم عامل و استفاده بعدی از آن‌ها برای تولید کانتینرهای جدید بسیار مفید است. دلیل اصلی این واقعیت است که آن‌ها بسیار سبک وزن هستند و همچنین استفاده و اشتراک گذاری آن‌ها آسان است.

امروزه کانتینرها اغلب در چارچوب‌های ریزسرویس رایانش ابری مورد استفاده قرار می گیرند، جایی که هر کانتینر می تواند به عنوان یک سرویس در نظر گرفته شود، با بقیه سرویس‌ها توسط شبکه ارتباط برقرار و مرتبط شود؛ بنابراین، این چارچوب‌ها اجازه می دهند تا هر عنصری مقیاس بندی شده و مستقل از سایرین استفاده شود، اما علاوه بر این، راه‌حل‌های مجازی سازی مبتنی بر کانتینر سبک وزن به عنوان فعال کننده‌های یک فناوری مجازی سازی کارآمدتر در محیط‌های اینترنت اشیا یا Fog مورد توجه قرار گرفته‌اند. در واقع، همان نمونه کانتینر مجازی سازی شده می تواند به طور موثر در گره‌های Fog و کاربر نهایی IoT و همچنین در فضای ابری اجرا شود.

کانتینرها را می توان بر روی تجهیزات محاسباتی بسیار محدود و نسبتاً ارزان، مانند Raspberry Pi اجرا کرد. این ویژگی قابلیت همکاری عرضی را در بین شبکه‌های مختلف ارائه می کند، دستگاه‌هایی را با منابع محدود محیط اینترنت اشیا/کاربران

نهایی یا Fog و همچنین دستگاه‌های ابری با قابلیت بالا یکپارچه می‌کند. ارائه خدمات مبتنی بر کانتینر می‌تواند مزایای متنوعی را برای شبکه‌های ابر-مه-اینترنت اشیاء ارائه دهد و به برنامه‌ها اجازه می‌دهد در مجموعه وسیعی از دستگاه‌ها با استقلال از سخت‌افزار زیربنایی اجرا شوند.

در مدیریت کانتینرها، عملکرد زمانبند یا کارگزارانی که مسئول تخصیص کانتینرها بین منابع هستند، اهمیت ویژه‌ای دارد. تعداد دستگاه‌های استفاده شده و عمر مفید تخمینی کانتینرها را می‌توان به‌عنوان فاکتورهای کلیدی برای زمان‌بندی کانتینر در نظر گرفت تا در فرآیند تصمیم‌گیری آن در نظر گرفته شود، برخلاف ماشین‌های مجازی/زمان‌بندی وظایف. زمان‌بندی کانتینرها در یک خوشه چندین هدف دارد: استفاده کارآمد از منابع خوشه، کار با محدودیت‌های مکان ارائه شده توسط کاربر، برنامه‌ریزی سریع برنامه‌ها به گونه‌ای که آن‌ها را به حالت انتظار هدایت نکند، ارائه درجه‌ای از "بی طرفی" یا تعادل بین منابع. در محیط ناهمگون اینترنت اشیاء و سیستم‌های مبتنی بر رایانش ابری، ماشین‌ها در مناطق مختلف قرار دارند و دارای قابلیت‌های پردازشی، ویژگی‌ها (تعداد هسته‌های پردازش، حافظه، پهنای باند و ...) و هزینه‌های متفاوتی هستند. علاوه بر این، انعطاف‌پذیری و پویایی ارائه شده توسط ابر و برخی چالش‌های مهم مانند کاهش مصرف انرژی و هزینه اقتصادی، پیچیدگی مشکل تخصیص منابع در محیط ابری را افزایش داده است. اگر P_i زمان اجرای وظیفه i باشد، آنگاه $L_i = \sum P_i$ زمان اتمام کار در ماشین مجازی i خواهد بود و در نتیجه $L_{max} = \max L_i$ را به عنوان زمان تکمیل در آن می‌نامیم. تمام ماشین‌های مجازی برخی از وظایف به صورت موازی با وظایف دیگر در چندین ماشین مجازی و برخی دیگر به صورت متوالی اجرا می‌شوند.

مدل فعلی شامل N ظرف مختلف است که در لایه مه به یکدیگر متصل می‌شوند. هر سلول مه یک مدیر سرور مه دارد که مسئولیت مدیریت درخواست‌های دریافتی را بر عهده دارد. مدیر سرور با استفاده از الگوریتم عقاب طلایی (GEO) منابع را در سلول‌های مه اختصاص می‌دهد. برای این منظور درخواست‌های کاربران محیط اینترنت اشیاء از طریق اپلیکیشن‌ها دریافت می‌شود و قرار دادن ریزسرویس‌ها بر اساس وضعیت منابع موجود انجام می‌شود. درخواست‌های دریافت شده توسط کاربران ممکن است مهلت‌های مختلفی داشته باشد؛ بنابراین برنامه زمان‌بندی باید به گونه‌ای باشد که اجرای کار در زمان مورد نظر به پایان برسد. ابتدا زمان سرویس هر درخواست را محاسبه می‌کنیم. تلورانس زمانی (حداکثر زمان قابل قبول) که تفاوت بین زمان فعلی و مهلت درخواست است، با فرمول زیر محاسبه می‌شود:

$$delay_i^T = (DL_i^T - C_i^T) \quad 1-0$$

کل زمان پاسخگویی به صورت زیر محاسبه می‌شود:

$$W_i = \frac{Q}{i} + \mu_i \quad 2-0$$

اولی کل زمان سپری شده در صف پردازش و دومی کل زمان خدمات مورد نیاز برای درخواست Req_i است. $delay_i^T$ حداکثر تاخیر مجاز برای درخواست طبق توافقات SLA است. برای احراز الزامات این موافقتنامه برای درخواست قی، باید شرایط زیر رعایت شود:

$$W_i < delay_i^T \quad 3-0$$

در رایانش ابری ماشین‌ها در نواحی مختلفی قرار دارند و دارای قابلیت‌های پردازشی، ویژگی‌ها (تعداد هسته‌های پردازشی، حافظه، پهنای باند و ...) و هزینه‌های متفاوتی هستند. علاوه بر این، انعطاف‌پذیری و پویایی ارائه شده توسط ابر و برخی چالش‌های مهم مانند کاهش مصرف انرژی و هزینه اقتصادی، پیچیدگی مشکل تخصیص منابع در محیط ابری را افزایش داده است.

طرح پایه

فناوری مجازی‌سازی کانتینر که توسط داکر ارائه شده است به دلیل مزایای سبک وزن بودن، استقرار سریع و قابلیت حمل آسان، به طور گسترده در صنعت استفاده شده است. تانگ و همکارانش [۲۱] سناریوهای برنامه‌های IoT مبتنی بر هوش مصنوعی را بر اساس فناوری کانتینر در یک محیط مشارکتی لبه ابری در نظر گرفته و یک الگوریتم زمان‌بندی کار مبتنی بر کانتینر را پیشنهاد کرده‌اند. با استفاده از استراتژی حریصانه آگاه از اولویت، یک الگوریتم زمان‌بندی جدید به نام PGT پیشنهاد شده است که از روش چند هدفه تکنیک اولویت بندی ترتیب با تعیین شباهت به راه حل ایده آل (TOPSIS) استفاده می‌کند.

کانتینرها در سرورهای ابری و سرورهای لبه در یک پلتفرم یکنواخت مدیریت می‌شوند و سرویس‌های کاربردی IoT در کانتینرها مستقر می‌شوند. کار با محدودیت ضرب الاجل کوچکتر به دلیل اولویت بالاتر زودتر زمان‌بندی می‌شود. سپس چندین شاخص به طور جامع مانند زمان پاسخگویی به کار، مصرف انرژی، هزینه اجرای کار، در نظر گرفته می‌شود تا راه حل بهینه برای اجرای کار پیدا شود. از طریق تغییر تعداد سرورهای لبه و تعداد وظایف، نتایج شبیه‌سازی در یک محیط مشارکتی لبه ابری نشان می‌دهد که رویکرد زمان‌بندی پیشنهادی در بهبود نرخ رضایت QoS، مصرف انرژی، هزینه جریمه و مجموع زمان نقض مهلت زمانی، عملکرد بهتری از چهار الگوریتم پایه دارد.

طرح پیشنهادی

اگر بتوانیم در زمان‌بندی کانتینرها توزیع آن‌ها روی ماشین‌های میزبان، دسترس پذیری و وضعیت منابع و ویژگی‌های ریزسرویس‌های خاص را لحاظ کنیم، یک زمان‌بندی آگاهانه انجام داده‌ایم که می‌تواند از نظر زمان اجرا و مصرف انرژی باعث بهبود عملکرد شود. هنگامی که استفاده از منابع لبه نمی‌تواند الزامات وظیفه را برآورده کند، کار برای اجرا در سرورهای ابری زمان‌بندی می‌شود تا به کیفیت سرویس (QoS) مورد نظر وظیفه پاسخ داده شود.

حال روش پیشنهادی را برای حل مساله زمان‌بندی وظایف مبتنی بر کانتینر در محیط مشارکتی محاسبات لبه و رایانش ابری توصیف می‌کنیم. وظایف با محدودیت‌های ضرب الاجل کوچکتر ابتدا به دلیل اولویت بالاتر زمان‌بندی می‌شوند به این معنا که ما به هر کار با توجه به مهلت تعیین شده یک مقدار اولویت اختصاص می‌دهیم. قوانین زمان‌بندی به شرح زیر توصیف می‌شوند:

۱. اگر سرویس‌های کاربردی به شکل کانتینر مستقر شده باشند که بتواند محدودیت ضرب الاجل یک وظیفه را برآورده کنند، هدف زمان‌بندی به حداقل رساندن هزینه اجرا و زمان پاسخگویی آن وظیفه است.
۲. اگر محدودیت ضرب الاجل کار توسط همه سرویس‌های کاربردی برآورده نشود، هدف زمان‌بندی به حداقل رساندن مصرف انرژی، هزینه اجرا و زمان پاسخگویی به آن وظیفه است.

در طرح پیشنهادی، زمان‌بندی مبتنی بر کانتینر جهت اجرای درخواست‌های کاربران، با استفاده از الگوریتم بهینه‌سازی عقاب طلایی (GEO) انجام می‌شود. در واقع هسته اصلی این الگوریتم برگرفته از هوش عقاب‌های طلایی در تنظیم سرعت شان در مراحل مختلف مسیر مار پیچش آن‌ها هنگام شکار است. در مراحل ابتدایی آن‌ها بیشتر تمایلشان به گشتن و جستجوی طعمه است و بیشتر در مراحل پایانی تمایل به حمله دارند، در واقع یک عقاب (عقاب) طلایی این دو مولفه را طوری تنظیم می‌کند تا بتواند در کمترین زمان ممکن بهترین طعمه را در ناحیه مورد نظر به دست آورد. این رفتار عقاب طلایی به صورت ریاضی مدل سازی شده است تا بتوان از آن برای اکتشاف (جستجوی طعمه) و بهره برداری (استثمار طعمه) برای یک روش بهینه

سازی سراسری، بهتر استفاده کرد. کارایی این الگوریتم بر روی ۳۳ بنچ مارک و یک آزمون مقیاس پذیری تست و تایید شده است. نتایج با شش الگوریتم معروف دیگر مقایسه شده است که گویای برتری الگوریتم GEO است. نتایج نشان دهنده این بوده است که این الگوریتم به خوبی می تواند بهینه سراسری را پیدا کند و مانع به دام افتادن در بهینه محلی شود [۲۲].

هر عقاب طلایی شکار خود را با پرواز در ارتفاع زیاد در قلمرو خود و در دایره های بزرگ در جستجوی طعمه آغاز می کند. هر وقت که طعمه مشاهده شد، شروع به حرکت در محیط یک دایره فرضی به مرکز طعمه می کند. عقاب طلایی مکان طعمه را به خاطر می سپارد اما به چرخیدن به دور آن ادامه می دهد. عقاب به تدریج ارتفاع خود را کاهش می دهد و همزمان با نزدیک شدن به طعمه شعاع دایره فرضی اطراف طعمه را کوچک و کوچکتر می کند. همزمان نواحی مجاور را نیز برای جایگزین های بهتر بررسی می کند. گاهی اوقات آن ها مکان بهترین طعمه هایی را که تاکنون پیدا کرده اند را با عقاب های دیگر نیز به اشتراک می گذارند. به عبارتی اگر عقاب طعمه یا مکان بهتری را پیدا نکند، به دور شکار فعلی بر روی دایره هایی که شعاع آن ها به مرور کوچک و کوچکتر می شود می چرخد و در نهایت به طعمه حمله می کند؛ اما اگر جایگزین بهتری پیدا کند بر روی یک دایره جدید در اطراف طعمه جدید پرواز میکند و طعمه قبلی را فراموش میکند. لازم به ذکر است که حمله نهایی در یک خط مستقیم انجام می شود. با این اوصاف با موارد گفته شده می توان ویژگی های بارز فرآیند شکار عقاب های طلایی را به صورت زیر خلاصه کرد:

۱. آن ها برای جستجوی طعمه یک مسیر مارپیچ و برای حمله به آن یک مسیر مستقیم را دنبال می کنند.
۲. در مراحل ابتدایی شکار آن ها بیشتر تمایلشان به گشتن و جستجوی طعمه است و سپس در مراحل پایانی به آرامی بیشتر متمایل به حمله به طعمه می شوند.
۳. آن ها در هر لحظه از پرواز تمایل به گشتن و حمله دارند.
۴. آن ها به دنبال اطلاعات عقاب های دیگر در مورد طعمه می گردند.

تعادل هوشمندانه عقاب های طلایی بین گشتن و حمله یک تجلی طبیعی از اکتشاف، بهره برداری و گذر از حالت اول به حالت دوم است (از گشتن به حمله). این کار راه را برای ابداع یک الگوریتم فرا ابتکاری هموار می کند.

هر عقاب طلایی بهترین مکانی را که تا به حال بازدید کرده به خاطر می سپارد و همزمان تمایلش به سمت حمله به طعمه و گشتن و جستجو برای غذای بهتر است. در شکل ۲ بردارهای *cruise* و *attack* در یک فضای دو بعدی نشان داده شده است. در هر تکرار، هر عقاب طلایی *i* به طور تصادفی طعمه عقاب طلایی دیگر را انتخاب می کند؛ و به دور بهترین مکانی که عقاب طلایی *f* تا به حال مشاهده کرده است می چرخد. علاوه بر این، عقاب طلایی *i* می تواند به دور مکانی که خود به خاطر سپرده است نیز بچرخد.

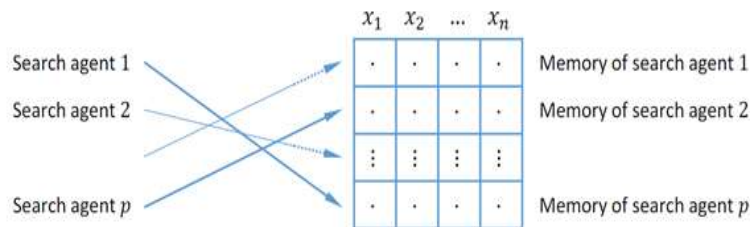


شکل ۲ - حرکات مارپیچی عقاب طلایی [۷۵]

هر عقاب طلایی در هر تکرار باید طعمه ای را انتخاب کند تا بتواند عملیات حمله و گشتن را انجام دهد. در GEO طعمه به صورت بهترین راه حلی که تا به حال توسط دسته ای از عقاب های طلایی پیدا شده است مدل سازی می شود. هر عقاب طلایی قادر است تا بهترین راه حلی که تا به حال پیدا کرده است را به خاطر بسپارد. در هر تکرار هر عامل جستجو یک طعمه

هدف را از بین طعمه‌های به خاطر سپرده شده توسط کل دسته عقاب‌ها انتخاب می‌کند. سپس برای هر عقاب طلایی بردارهای attack و cruise متناسب با طعمه انتخاب شده محاسبه می‌شوند. اگر مکان جدیدی که توسط بردارهای attack و cruise محاسبه شده است بهتر از مکان قبلی که در حافظه عقاب بوده است باشد حافظه آن در روز رسانی می‌شود. به طور کلی استراتژی انتخاب طعمه نقش مهمی را در این الگوریتم بازی می‌کند

انتخاب طعمه نیز می‌تواند به سادگی صورت گیرد. به گونه‌ای که هر عقاب طلایی فقط طعمه‌ای را انتخاب می‌کند که در خاطره اوست. برای اینکه عقاب‌های طلایی به کاوش بهتر در مناظر بپردازند، یک طرح نقشه برداری تصادفی یک به یک در این مقاله ارائه شده است؛ که در آن هر عقاب طلایی به طور تصادفی طعمه خود را در تکرار فعلی از حافظه هر یک از اعضای دیگر گله انتخاب می‌کند. باید در نظر داشته باشیم که طعمه انتخابی لزوماً دورترین یا نزدیک‌ترین طعمه نیست. در این طرح پیشنهادی، هر طعمه در حافظه تنها به یک عقاب طلایی اختصاص داده می‌شود. سپس هر عقاب عملیات حمله و کروزر را روی طعمه انتخابی انجام می‌دهد. شکل ۳ نشان می‌دهد که هر عامل جستجو تنها می‌تواند به یکی از مکان‌های حافظه متعلق به عامل جستجوی دیگر حمله کند.



شکل ۳- نگاهت یک به یک در انتخاب طعمه در حافظه عامل جستجو [۷۵]

حمله می‌تواند به صورت برداری که از موقعیت فعلی عقاب طلایی شروع شده و به مکان طعمه در حافظه عقاب ختم می‌شود، مدل شود. بردار حمله برای عقاب طلایی i می‌تواند به صورت رابطه ۱ محاسبه شود:

$$\vec{A}_i = \vec{X}_f^* - \vec{X}_i \quad 4-0$$

در این فرمول A_i بردار حمله ایگل i است؛ و X_f بهترین مکان (شکار) است که تا به حال توسط eagle f دیده شده است و X_i موقعیت کنونی eagle i است. از آنجایی که بردار حمله مجموعه‌ای از عقاب‌های طلایی را به بهترین مکان مشاهده شده هدایت می‌کند، می‌تواند به خوبی گام استخراج را در الگوریتم GEO نشان دهد. بردار کاوش بر اساس بردار حمله محاسبه می‌شود. درواقع بردار اکتشاف بردار مماس بر دایره و عمود بر بردار حمله می‌باشد؛ که در شکل ۲ نیز نشان داده شده است. کروزر را می‌توان به عنوان سرعت خطی عقاب طلایی نسبت به طعمه نیز در نظر گرفت. بردار کروزر در n بعد داخل ابر صفحه مماس بر دایره است؛ بنابراین، برای محاسبه این بردار، ابتدا باید معادله ابر صفحه مماس را محاسبه کنیم. معادله یک ابر صفحه در ابعاد n را می‌توان با یک نقطه دلخواه و یک بردار عمود بر آن ابر صفحه تعیین کرد که به آن بردار عادی ابر صفحه می‌گویند. معادله زیر شکل اسکالر معادله ابر صفحه را در فضای n بعدی نشان می‌دهد.

$$h_1x_1 + h_2x_2 + \dots + h_nx_n = d \Rightarrow \sum_{j=1}^n h_jx_j = d \quad 5-0$$

که در آن بردار $\vec{H} = [h_1, h_2, \dots, h_n]$ یک نقطه دلخواه در ابر صفحه است که در نهایت به شکل d است اگر X_i را که محل عقاب i است، به عنوان یک نقطه دلخواه در ابر صفحه و A_i در نظر بگیریم، که بردار حمله است، به عنوان بردار معمولی ابر صفحه، می‌توانیم ابر صفحه را به صورت بردار کروزر برای عقاب طلایی i در تکرار t مطابق معادله زیر نشان دهیم.

$$\sum_{j=1}^n a_j x_j = \sum_{j=1}^n a'_j x'_j$$

۶-۰

که در آن برد $\vec{A}_i = [a_1, a_2, \dots, a_n]$ بردار متغیرهای تصمیم/ طراحی است و $X' = [x'_1, x'_2, \dots, x'_n]$ مکان طعمه انتخاب شده است.

حال می توان ابرصفحه **cruise** برای عقاب i در تکرار t را محاسبه کرد. به عبارتی زمان آن رسیده است که بردار **cruise** برای این عقاب طلایی را در این ابر صفحه پیدا کنیم. یک عقاب طلایی می تواند هر نقطه مقصدی را بر روی ابرصفحه **cruise** انتخاب کند. برای یافتن یک بردار تصادفی بر روی ابرصفحه **cruise** ابتدا باید یک نقطه مقصد تصادفی C به غیر از آن نقطه ای که از قبل داشتیم (مکان فعلی عقاب i) پیدا کنیم. باید به یاد داشته باشیم که نقطه شروع بردار **cruise** مکان فعلی شهاب طلایی i است. از آنجایی که ابرصفحه ها یک بعد کوچکتر از فضای اطراف خود هستند، نمی توانیم به سادگی یک نقطه تصادفی $n-1$ * در آن ایجاد کنیم؛ به عبارت دیگر، یک نقطه تصادفی ساده در فضای n بعدی تضمین نمی کند که در هایپرپلان $n-1$ قرار دارد. نقطه جدیدی که در نهمین ابر کروز قرار دارد، $n-1$ درجه آزادی دارد. این بدان معنی است که ابعاد $n-1$ را می توان آزادانه انتخاب کرد؛ اما همانطور که در معادله ۳-۵ نشان داده شده است، معادله هایپرپلان نشان دهنده آخرین بعد است. بعد آخر باید طوری انتخاب شود که معادله هایپرپلان برقرار باشد؛ بنابراین ما $n-1$ متغیر رایگان و یک متغیر ثابت داریم. شایان ذکر است که بردار کروز جمعیتی از عقاب های طلایی را به مناطقی غیر از مواردی که حفظ کرده اند و در حافظه خود جذب کرده اند، جذب می کند؛ بنابراین بر مرحله اکتشاف در **GEO** تاکید می کند.

جابجایی عقاب های طلایی شامل حمله و بردار است؛ که بردار گام برای عقاب طلایی i در تکرار t ام به صورت رابطه زیر نشان داده میشود که در این معادله ضریب حمله در هر تکرار t و ضریب **cruise** در هر تکرار t است و تنظیم می کند p_c^t عقاب های طلایی چگونه تحت تاثیر حمله و **cruise** قرار می گیرند.

$$\Delta x_i = \vec{r}_1 p_a \frac{\vec{A}_i}{\|\vec{A}_i\|} + \vec{r}_2 p_c \frac{\vec{C}_i}{\|\vec{C}_i\|} \quad 7-0$$

بردارهای r_1 و r_2 بردارهای تصادفی هستند که عناصر آن ها در بازه صفر و یک قرار دارند. در مورد p_a و p_c بعداً صحبت می شود؛ و نرم اقلیدسی بردارهای \vec{A}_i و \vec{C}_i هستند که با استفاده از معادله زیر محاسبه می شوند.

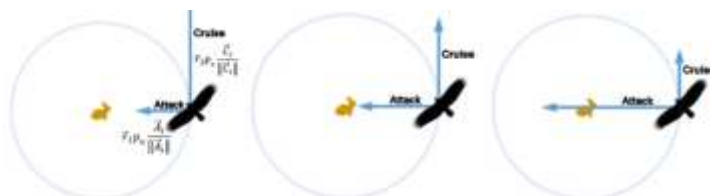
$$\|\vec{A}_i\| = \sqrt{\sum_{j=1}^n a_j^2}, \quad \|\vec{C}_i\| = \sqrt{\sum_{j=1}^n c_j^2} \quad 8-0$$

موقعیت عقاب ها در تکرار $t+1$ به راحتی با افزودن بردار گام به موقعیت فعلی در تکرار t محاسبه می شود:

$x^{t+1} = x^t + \Delta x_i^t$	۹-۰
--------------------------------	-----

اگر تناسب (fitness) موقعیت جدید عقاب طلایی i بهتر از موقعیتی باشد که در حافظه آن قرار دارد، حافظه این عقاب طلایی با موقعیت جدید به روز رسانی می شود. در غیر این صورت حافظه به صورت دست نخورده باقی می ماند، اما عقاب در موقعیت جدید ساکن می شود. در تکرار جدید هر عقاب طلایی به صورت تصادفی یک عقاب طلایی را از بین جمعیت عقاب ها انتخاب می کند تا در اطراف بهترین مکانی که بازدید کرده است بچرخد و حلقه بزند. بردار **attack** و **cruise** را محاسبه کند و در نهایت بردار گام و موقعیت جدید برای تکرار بعدی را محاسبه کند. این حلقه تا زمانی که یکی از شرط های خاتمه برآورده شود اجرا می شود.

همانطور که قبلا بیان شد عقاب های طلایی در مراحل اولیه پرواز شکاری تمایل بیشتری به گشتن یا cruise و در مراحل پایانی تمایل بیشتری به حمله نشان می دهند؛ که این مطابق با اکتشاف بیشتر در تکرارهای اولیه و استعمار و بهره برداری بیشتر در تکرار های پایانی در بهینه سازی پیشنهادی است. شکل نحوه تغییر حمله و کروز را نشان می دهد. الگوریتم GEO برای انتقال از حالت کاوش به حالت بهره برداری از p_a و کامپیوتر استفاده می کند. الگوریتم با مقدار p_a کم و p_c زیاد شروع می شود. مقدار اولیه و نهایی هر دو پارامتر توسط کاربر تعریف می شود. مقادیر میانی را نیز می توان با استفاده از انتقال خطی نشان داده شده در رابطه زیر محاسبه کرد.



شکل ۴- انتقال از حالت اکتشاف به حالت حمله در روش عقاب طلایی [۷۵]

$$\begin{cases} p_a = p_a^0 + \frac{t}{T} |p_a^T - p_a^0| \\ p_c = p_c^0 - \frac{t}{T} |p_c^T - p_c^0| \end{cases} \quad 100$$

در این معادله t تکرار فعلی، T حداکثر تکرار و به ترتیب مقادیر اولیه و نهایی برای تمایل به حمله و پارامترهای و مقادیر اولیه و نهایی برای p_c ، p_a^T ، p_c^T ، p_a^0 ، p_c^0 cruise (pc) هستند. آزمایش های انجام شده این مقاله نشان می دهند که مقدار p_a در تکرار اول بر روی 0.5 تنظیم شده است و به صورت خطی کاهش می یابد تا در آخرین تکرار به 2 برسد. در مورد p_c هم همین طور، در تکرار اول با مقدار 1 شروع می شود و به صورت خطی کاهش می یابد تا در تکرار آخر به 0.5 برسد. شایان ذکر است که معادله بالا پارامترها را به صورت خطی تغییر می دهد اما به طور کلی می توان آن ها را به صورت لگاریتمی یا با استفاده از هر تابع دیگر تغییر داد. شبه کد پیاده سازی GEO در الگوریتم زیر نشان داده شده است.

```

Algorithm 1. Pseudo-code of GEO
Initialize the population of golden eagles
Evaluate fitness function
Initialize population memory
Initialize  $p_a$  and  $p_c$ 
for each iteration  $t$ 
    Update  $p_a$  and  $p_c$  (Equation (9))
    for each golden eagle  $i$ 
        Randomly select a prey from the population's memory
        Calculate attack vector  $\vec{A}$  (Equation (1))
        if attack vector's length is not equal to zero
            Calculate cruise vector  $\vec{C}$  (Equations (2)-(5))
            Calculate step vector  $\Delta x$  (Equations (6)-(8))
            Update position (Equation (8))
            Evaluate fitness function for the new position
            if fitness is better than the fitness of the position in eagle  $i$ 's memory
                Replace the new position with the position in eagle  $i$ 's memory
            end
        end
    end
end
end
    
```

شکل ۴- شمای کلی یک راه حل در روش عقاب طلایی

شمای کلی یک راه حل در روش عقاب طلایی بصورت شکل ۴ می باشد. در این تصویر می بینیم که امکان ایجاد ترکیب های مختلفی از تخصیص کانتینرها به درخواست های کاربران وجود دارد.

جدول ۱- ساختار یک نمونه راه حل در طرح پیشنهادی

Task	T1	T2	T3	T4	T5	T6	T7
Container	C2	C1	C2	C3	C8	C1	C6

همانگونه که در جدول ۱ می بینیم اندازه یک راه حل برابر تعداد وظایف درخواستی می باشد. مقادیر خانه های ماتریس، شماره کانتینرهای تخصیص یافته برای انجام وظیفه i می باشد. برای ارزیابی هر راه حل، یک تابع برازش (هدف) در طول بهینه سازی استفاده می شود. تلاش ما در طرح کنونی این است که زمان پاسخگویی، مصرف انرژی را به حداقل برسانیم و زمان تخطی از توافقات سطح سرویس (SLA) در زمان بندی وظایف بر روی کانتینرها پیش رویم. منظور از زمان تخطی از توافقات، تفاوت زمانی ضرب العجل یک وظیفه به عنوان مهلت انجام آن درخواست و زمان اجرای وظیفه پس از زمان بندی بر روی یک کانتینر می باشد.

$$Violation_SLA = \sum_{i=1}^T (finish(i) - deadline(i)) \quad 11-0$$

بطوریکه $Violation_SLA$ مجموع زمان تخطی از مهلت میکروسرویس ها، $finish(i)$ زمان تکمیل وظیفه i بر روی کانتینر اختصاص داده شده، و $deadline(i)$ ضرب العجل اجرای وظیفه i است.

میزان انرژی مصرفی حاصل از اجرای یک سرویس در یک کانتینر به ماشین میزبان آن کانتینر و حجم درخواست تخصیص داده شده به کانتینر بستگی دارد.

$$E_j = HE_j + U_j \cdot load_j \quad 12-0$$

بطوریکه U_j هزینه واحد مصرف انرژی در کانتینر j $load_j(t)$ میزان درخواست های ورودی به این کانتینر در آن برهه زمانی می باشد. زمان پاسخ را بصورت میانگین مدت زمان اجرای وظایف روی کانتینر بر اساس زمان بندی ارائه شده در نظر می گیریم.

$$Resp = \frac{1}{T} \sum_{i=1}^T run(i) \quad 13-0$$

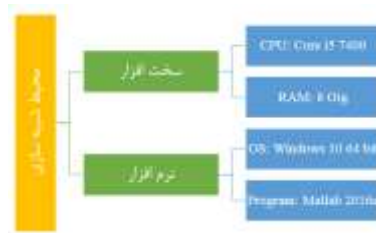
بنابراین تابع برازش نهایی بصورت زیر تعریف می گردد

$$fitness(sol_i) = \frac{C_1}{Violation_SLA} + \frac{C_2}{\sum_{j \in Tsk} E_j} + \frac{C_3}{Resp} \quad 14-0$$

$D(sol_i)$ میانگین زمان پاسخ در زمان بندی درخواست های کاربران، E_j انرژی مصرفی اجرای وظیفه j و C_1 ضرایب ا فاده شده برای وزن دهی به پارامترهای موثر در زمان بندی بهینه است. با اجرای طرح پیشنهادی، پارامترهای زمان پاسخگویی و مصرف انرژی کاهش و همچنین کیفیت خدمات (تعداد وظایف انجام شده در مهلت تعیین شده) افزایش می یابد.

روش تجزیه و تحلیل

برای شبیه سازی طرح پیشنهادی از نرم افزار متلب ۲۰۱۶ استفاده شد. مشخصات سیستم برای پیاده سازی روش پیشنهادی در شکل ۵ نشان داده شده است.



شکل ۵- مشخصات محیط شبیه سازی

پارامترهای در طرح پیشنهادی

طرح پیشنهادی پایان نامه یک رویکرد جدید برای مجازی سازی مبتنی بر کانتینر و زمان بندی ریزسرویس های ابر است. در فرآیند شبیه سازی پارامترهایی برای طرح پیشنهادی وجود دارد که مقدار این پارامترها در نتایج و عملکرد طرح تاثیر دارند. جدول ۲ این پارامترها را نشان می دهد.

جدول ۲- معرفی متغیرهای شبیه سازی

مقدار	متغیر
۱۰	تعداد کانتینر
۳۰۰، ۲۵۰، ۲۰۰، ۱۵۰، ۱۰۰	تعداد وظایف
100	زمان شبیه سازی (راند)
۶	تعداد میزبان ها
۲۰۰	تعداد جمعیت اولیه
۱۰۰ نسل	دفعات تکرار بهینه سازی

شاخص های ارزیابی

در این پایان نامه برای سنجش روش پیشنهادی از شش شاخص مطرح در رایانش ابری استفاده شده است. این شاخص ها زمان پاسخگویی، زمان اتمام وظایف، انرژی مصرفی، تخطی از زمان تعیین شده، تعداد برنامه های اجرا شده در خارج از زمان تعیین شده، سر بار محاسباتی (زمان زمان بندی) هستند. جدول ۳ این شاخص ها را معرفی می کند.

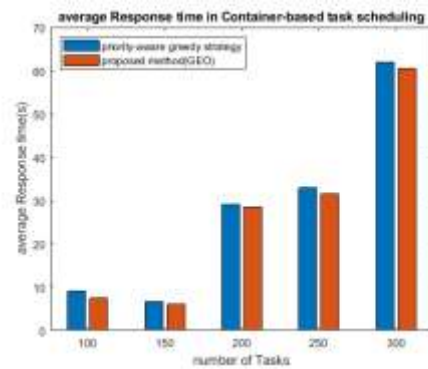
جدول ۳- معرفی شاخص های ارزیابی طرح پیشنهادی

توضیحات شاخص	شاخص
این شاخص نشان می دهد که برای اتمام کارها/وظایف چه مقدار زمان صرف می شود.	زمان اتمام کارها
این معیار بیانگر میزان انرژی مصرفی شبکه در هر دور اجرا می باشد.	انرژی مصرفی
این شاخص مجموع زمان استقرار و متوسط زمان اجرا خواهد بود که زمان سرویس یا زمان پاسخگویی را مشخص می کند.	زمان پاسخگویی
هر فرآیند دارای محدودیت زمانی برای اجرا می باشد. این مهلت حداکثر زمانی است که اجرای یک فرآیند می تواند به تعویق بیفتد. این اندازه گیری نرخ تخلف فرآیندها را از زمان مهلت اجرا مدل می کند.	مهلت زمانی انجام وظایف
این معیار تعداد برنامه هایی که زمان اجرای آن ها از مهلت مقرر تخطی کرده است؛ را نشان می دهد. این شاخص بر حسب عدد یا درصد نشان داده می شود.	تعداد برنامه های اجرا شده خارج مهلت زمانی

این معیار بیانگر زمان انجام محاسبات در هر روش می باشد که بر حسب ثانیه محاسبه می گردد. در روش پیشنهادی این شاخص معرف زمان لازم برای زمان بندی وظایف می باشد.	پنجیدگی محاسباتی
---	------------------

زمان پاسخگویی

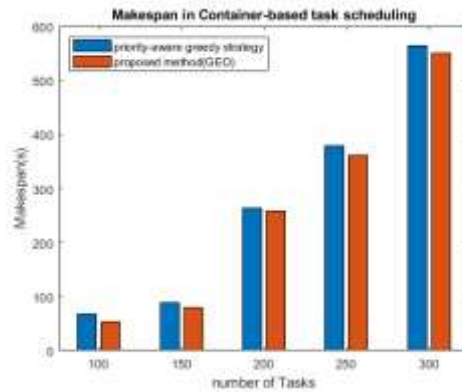
بررسی کارایی طرح پایه و طرح پیشنهادی با شاخص زمان پاسخگویی نشان می دهد که به صورت میانگین زمان پاسخگویی طرح پیشنهادی که یک رویکرد مبتنی بر الگوریتم بهینه سازی عقاب طلایی می باشد؛ نسبت به طرح پایه کمتر می باشد. نمودار ۱ نتایج این بررسی را نشان می دهد. در شاخص زمان پاسخگویی، الگوریتم و روشی مطلوب بحساب می آید که زمان پاسخگویی کمتری نسبت به دیگر روش ها داشته باشد؛ اما تفسیر نتایج بیانگر آن است که در شاخص زمان پاسخگویی، طرح پیشنهادی پایان نامه به صورت میانگین ۸ درصد نسبت به طرح پایه که یک رویکرد مبتنی بر استراتژی حریرانه و آگاه از اولویت می باشد؛ بهبود یافته است.



نمودار ۱- ارزیابی کارایی طرح ها در شاخص زمان پاسخگویی

زمان اتمام کارها

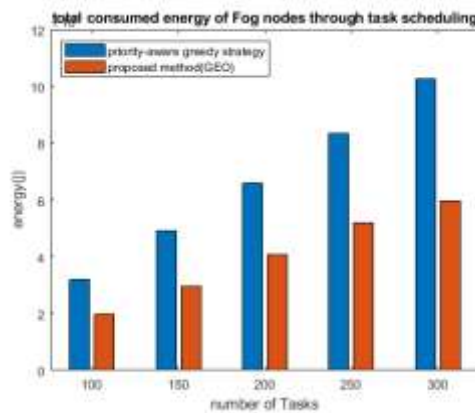
بررسی کارایی طرح پایه و طرح پیشنهادی با شاخص زمان اتمام کارها نشان می دهد که به صورت میانگین زمان اتمام کارهای طرح پیشنهادی که یک رویکرد مبتنی بر الگوریتم بهینه سازی عقاب طلایی می باشد؛ نسبت به طرح پایه کمتر می باشد. نمودار (۲-۴) نتایج این بررسی را نشان می دهد. در شاخص زمان اتمام کارها، الگوریتم و روشی مطلوب بحساب می آید که زمان اتمام کارهای کمتری نسبت به دیگر روش ها داشته باشد؛ اما تفسیر نتایج بیانگر آن است که در شاخص زمان اتمام کارها، طرح پیشنهادی پایان نامه به صورت میانگین ۷ درصد نسبت به طرح پایه که یک رویکرد مبتنی بر استراتژی حریرانه و آگاه از اولویت می باشد؛ بهبود یافته است.



نمودار ۲- ارزیابی کارایی طرح ها در شاخص زمان اتمام کارها

انرژی مصرفی

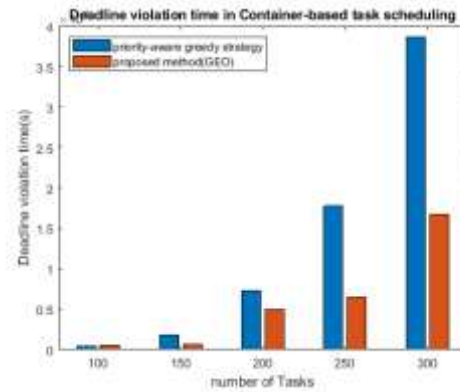
بررسی کارایی طرح پایه و طرح پیشنهادی با شاخص انرژی مصرفی نشان می دهد که به صورت میانگین انرژی مصرفی طرح پیشنهادی که یک رویکرد مبتنی بر الگوریتم بهینه سازی عقاب طلایی می باشد؛ نسبت به طرح پایه کمتر می باشد. نمودار ۳ نتایج این بررسی را نشان می دهد. در شاخص انرژی مصرفی شبکه، الگوریتم و روشی مطلوب بحساب می آید که انرژی مصرفی کمتری نسبت به دیگر روش ها داشته باشد؛ اما تفسیر نتایج بیانگر آن است که در شاخص انرژی مصرفی، طرح پیشنهادی پایان نامه به صورت میانگین ۲۷ درصد نسبت به طرح پایه که یک رویکرد مبتنی بر استراتژی حریرانه و آگاه از اولویت می باشد؛ بهبود یافته است.



نمودار ۳- ارزیابی کارایی طرح ها در شاخص انرژی مصرفی

تخطی از زمان تعیین شده

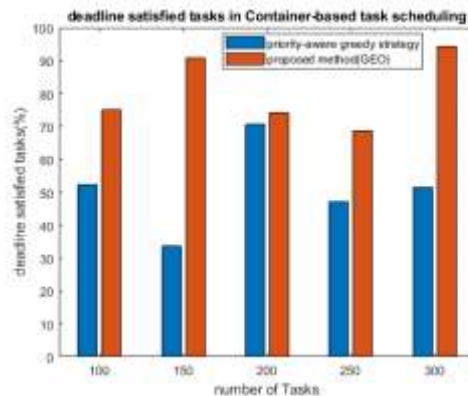
بررسی کارایی طرح پایه و طرح پیشنهادی با شاخص تخطی از زمان تعیین شده نشان می دهد که به صورت میانگین تخطی از زمان تعیین شده طرح پیشنهادی که یک رویکرد مبتنی بر الگوریتم بهینه سازی عقاب طلایی می باشد؛ نسبت به طرح پایه کمتر می باشد. نمودار ۴ نتایج این بررسی را نشان می دهد. در شاخص تخطی از زمان تعیین شده، الگوریتم و روشی مطلوب بحساب می آید که تخطی از زمان تعیین شده کمتری نسبت به دیگر روش ها داشته باشد؛ اما تفسیر نتایج بیانگر آن است که در شاخص تخطی از زمان تعیین شده، طرح پیشنهادی پایان نامه به صورت میانگین ۲۴ درصد نسبت به طرح پایه که یک رویکرد مبتنی بر استراتژی حریرانه و آگاه از اولویت می باشد؛ بهبود یافته است.



نمودار ۴- ارزیابی کارایی طرح ها در شاخص تخطی از زمان تعیین شده

تعداد برنامه های اجرا شده در خارج از زمان ضرب الاجل

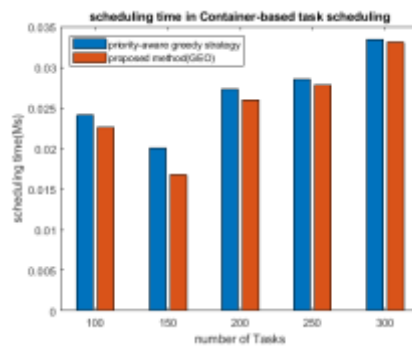
بررسی کارایی طرح پایه و طرح پیشنهادی با شاخص تعداد برنامه های اجرا شده در خارج از زمان ضرب الاجل نشان می دهد که به صورت میانگین تعداد برنامه های اجرا شده در خارج از زمان ضرب الاجل برای طرح پیشنهادی که یک رویکرد مبتنی بر الگوریتم بهینه سازی عقاب طلایی می باشد؛ نسبت به طرح پایه کمتر می باشد. نمودار ۵ نتایج این بررسی را نشان می دهد. در این شاخص، الگوریتم و روشی مطلوب بحساب می آید که داد برنامه های اجرا شده در خارج از زمان ضرب الاجل کمتری نسبت به دیگر روش ها داشته باشد؛ اما تفسیر نتایج بیانگر آن است که در شاخص داد برنامه های اجرا شده در خارج از زمان ضرب الاجل، طرح پیشنهادی پایان نامه به صورت میانگین ۲۲ درصد نسبت به طرح پایه که یک رویکرد مبتنی بر استراتژی حریمانه و آگاه از اولویت می باشد؛ بهبود یافته است.



نمودار ۵- ارزیابی کارایی طرح ها در شاخص تعداد برنامه های اجرا شده در خارج از زمان ضرب الاجل

پیچیدگی محاسباتی (زمان زمان بندی)

بررسی کارایی طرح پایه و طرح پیشنهادی با شاخص پیچیدگی محاسباتی (زمان زمان بندی) نشان می دهد که به صورت میانگین زمان مورد نیاز برای زمان بندی کارها در طرح پیشنهادی که یک رویکرد مبتنی بر الگوریتم بهینه سازی عقاب طلایی می باشد؛ نسبت به طرح پایه کمتر می باشد. نمودار ۶ نتایج این بررسی را نشان می دهد. در این شاخص، الگوریتم و روشی مطلوب بحساب می آید که پیچیدگی محاسباتی (زمان زمان بندی) کمتری نسبت به دیگر روش ها داشته باشد؛ اما تفسیر نتایج بیانگر آن است که در شاخص پیچیدگی محاسباتی (زمان زمان بندی)، طرح پیشنهادی پایان نامه به صورت میانگین ۷ درصد نسبت به طرح پایه که یک رویکرد مبتنی بر استراتژی حریمانه و آگاه از اولویت می باشد؛ بهبود یافته است.



نمودار ۶- ارزیابی کارایی طرح ها در شاخص پیچیدگی محاسباتی (زمان زمان بندی)

بحث و نتیجه گیری

در این مقاله برای ارزیابی روش پیشنهادی از شش شاخص مطرح در رایانش ابری استفاده شده است. این شاخص ها زمان پاسخگویی، زمان اتمام وظایف، انرژی مصرفی، تخطی از زمان تعیین شده، تعداد برنامه های اجرا شده در خارج از زمان تعیین شده، سر بار محاسباتی (زمان زمان بندی) هستند. بررسی آزمایش های مختلف گویای آن است که:

۱. در شاخص زمان پاسخگویی، طرح پیشنهادی مقاله به صورت میانگین ۸ درصد نسبت به طرح پایه که یک رویکرد مبتنی بر استراتژی حریصانه و آگاه از اولویت می باشد؛ بهبود یافته است.
۲. در شاخص زمان اتمام کارها، طرح پیشنهادی مقاله به صورت میانگین ۷ درصد نسبت به طرح پایه بهبود یافته است.
۳. در شاخص انرژی مصرفی، طرح پیشنهادی مقاله به صورت میانگین ۲۷ درصد نسبت به طرح پایه بهبود یافته است.
۴. در شاخص تخطی از زمان تعیین شده، طرح پیشنهادی مقاله به صورت میانگین ۲۴ درصد نسبت به طرح پایه بهبود یافته است.
۵. در شاخص برنامه های اجرا شده در خارج از زمان ضرب الاجل، طرح پیشنهادی مقاله به صورت میانگین ۲۲ درصد نسبت به طرح پایه بهبود یافته است.
۶. در شاخص پیچیدگی محاسباتی (زمان زمان بندی)، طرح پیشنهادی مقاله به صورت میانگین ۷ درصد نسبت به طرح پایه بهبود یافته است.

منابع

1. M.-G. Avram, "Advantages and challenges of adopting cloud computing from an enterprise perspective," *Procedia Technology*, vol. ۱۲, pp. ۵۳۴-۵۲۹, ۲۰۱۴
2. S. Suo, *Cloud implementation in organizations: Critical success factors, challenges, and impacts on the it function*. The Pennsylvania State University, ۲۰۱۳
3. Lin and N.-C. Chen, "Cloud computing as an innovation: Perception, attitude, and adoption," *International journal of information management*, vol. ۳۲, no. ۶, pp. ۵۴۰-۵۳۳, ۲۰۱۲
4. N. Sultan, "Cloud computing for education: A new dawn?," *International Journal of Information Management*, vol. ۳۰, no. ۲, pp. ۱۱۶-۱۰۹, ۲۰۱۰
5. Y. Zhang, X. Lan, J. Ren, and L. Cai, "Efficient computing resource sharing for mobile edge-cloud computing networks," *IEEE/ACM Transactions on Networking*, vol. ۲۸, no. ۳, pp. ۱۲۴۰-۱۲۲۷, ۲۰۲۰
6. 34J. Sha, A. G. Ebadi, D. Mavaluru, M. Alshehri, O. Alfarradj, and L. Rajabion, "A

method for virtual machine migration in cloud computing using a collective behavior-based metaheuristics algorithm," *Concurrency and Computation: Practice and Experience*, vol. ۳۲, no. ۲, p. e۵۴۴۱, .۲۰۲۰

7. K. Baalamurugan and S. V. Bhanu, "A multi-objective krill herd algorithm for virtual machine placement in cloud computing," *The Journal of Supercomputing*, vol. ۷۶, no. ۶, pp. ۴۵۴۲-۴۵۲۵, .۲۰۲۰

8. M. Choi, J. Park, and Y.-S. Jeong, "Mobile cloud computing framework for a pervasive and ubiquitous environment," *The Journal of Supercomputing*, vol. ۶۴, pp. ۳۵۶-۳۳۱, .۲۰۱۳

9. J. Sha, A. G. Ebadi, D. Mavaluru, M. Alshehri, O. Alfarraj, and L. Rajabion, "A method for virtual machine migration in cloud computing using a collective behavior-based metaheuristics algorithm," *Concurrency and Computation: Practice and Experience*, vol. ۳۲, no. ۲, p. e۵۴۴۱, .۲۰۲۰

10. Genco and S. Sorce, *Pervasive systems and ubiquitous computing*. Wit Press, .۲۰۱۰

11. M. Hwang, D. Choi, and P. Kim, "Least Slack Time Rate first: an Efficient Scheduling Algorithm for Pervasive Computing Environment," *J. Univers. Comput. Sci.*, vol. ۱۷, no. ۶, pp. ۹۲۵-۹۱۲, ...۲۰۱۱

12. M. Masdari, S. ValiKardan, Z. Shahi, and S. I. Azar, "Towards workflow scheduling in cloud computing: a comprehensive analysis," *Journal of Network and Computer Applications*, vol. ۶۶, pp. ۸۲-۶۴, .۲۰۱۶

13. Ahmad, M. G. AlFailakawi, A. AlMutawa, and L. Als Salman, "Container scheduling techniques: A survey and assessment," *Journal of King Saud University-Computer and Information Sciences*, vol. ۳۴, no. ۷, pp. ۳۹۴۷-۳۹۳۴, .۲۰۲۲

14. D. Kovachev, T. Yu, and R. Klamma, "Adaptive computation offloading from mobile devices into the cloud," in *۲۰۱۲ IEEE ۱۰th International Symposium on Parallel and Distributed Processing with Applications*, ۲۰۱۲, pp. ۷۹۱-۷۸۴: IEEE.

15. M. Shiraz and A. Gani, "A lightweight active service migration framework for computational offloading in mobile cloud computing," *The Journal of Supercomputing*, vol. ۶۸, pp. ۹۹۵-۹۷۸, .۲۰۱۴

16. R. Pérez de Prado, S. García-Galán, J. E. Muñoz-Expósito, A. Marchewka, and N. Ruiz-Reyes, "Smart containers schedulers for microservices provision in cloud-fog-IoT networks. Challenges and opportunities," *Sensors*, vol. ۲۰, no. ۶, p. ۱۷۱۴, .۲۰۲۰

17. R. Roostaei and Z. Movahedi, "Mobility and context-aware offloading in mobile cloud computing," in *۲۰۱۶ Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCOM/IoP/SmartWorld)*, ۲۰۱۶, pp. ۱۱۴۸-۱۱۴۴: IEEE.

18. S. E. Mahmoodi, R. Uma, and K. Subbalakshmi, "Optimal joint scheduling and cloud offloading for mobile applications," *IEEE Transactions on Cloud Computing*, vol. ۷, no. ۲, pp. ۳۱۳-۳۰۱, .۲۰۱۶

19. M. G. Xavier, M. V. Neves, F. D. Rossi, T. C. Ferreto, T. Lange, and C. A. De Rose, "Performance evaluation of container-based virtualization for high performance computing

environments," in *21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, ۲۰۱۳, pp. ۲۴۰-۲۳۳: IEEE.

20. H. Tout, C. Talhi, N. Kara, and A. Mourad, "Selective mobile cloud offloading to augment multi-persona performance and viability," *IEEE Transactions on Cloud Computing*, vol. ۷, no. ۲, pp. ۳۲۸-۳۱۴, ۲۰۱۶
21. C. Fehling *et al.*, "Cloud computing fundamentals," *Cloud Computing Patterns: Fundamentals to Design, Build, and Manage Cloud Applications*, pp. ۷۸-۲۱, ۲۰۱۴
22. L. Yang, J. Cao, S. Tang, D. Han, and N. Suri, "Run time application repartitioning in dynamic mobile cloud environments," *IEEE Transactions on Cloud Computing*, vol. ۴, no. ۳, pp. ۳۴۸-۳۳۶, ۲۰۱۴