

ارائه یک الگوریتم بهینه برای زمان بندی منابع و پارتیشن بندی کد در محاسبات ابری تلفن همراه*

راضیه قاسمی^۱

فرزانه فاموری^۲

چکیده

با تکنولوژی مجازی سازی، مراکز داده ابری امروزی انعطاف پذیرتر و امن تر می شود و بر اساس تقاضا تخصیص می یابد. یک فناوری کلیدی که نقش مهمی در مراکز داده ابری بازی می کند، برنامه زمان بندی منابع است. در این مقاله برای حل مشکلات موجود در این حوزه با استفاده از الگوریتم تکاملی ازدحام ذرات برای کوچک کردن حیطه اهداف چندگانه به اندازه های مناسب، راهبرد جایابی نزدیک به بهینه ارائه شده است. روش جایابی مبتنی بر الگوریتم بهینه سازی ازدحام ذرات، به دلیل افزایش در قابلیت محاسباتی پردازشگرها در پنج سال گذشته می تواند به عنوان یک روش جایابی زمان واقعی عمل کند. این جایابی یک روش جستجو بوده که در آن شایستگی به صورتی پویا بر مبنای واریانس های مقادیر تناسب در هر نسل تغییر می کنند. این روش مهاجرت و جایابی، به حداقل سازی زمان تکمیل ماشین های مجازی نیز توجه می کند. برای ارزیابی روش پیشنهادی، نتایج حاصله طی معیارهای متفاوت کیفیتی و از جنبه های متفاوت و بر اساس تغییر در پارامترهای مختلف اجرایی مورد مقایسه و بررسی قرار گرفت. کارایی روش معرفی شده در مقایسه با سایر روش های موجود در این زمینه مقایسه شد که نتایج بیانگر کیفیت بالای روش پیشنهادی می باشد.

واژه های کلیدی: بهینه سازی ازدحام ذرات، پارتیشن بندی کد، زمان بندی منابع، ماشین مجازی، محاسبات ابری تلفن همراه.

* تاریخ دریافت: ۱۳۹۸/۰۲/۱۰؛ تاریخ پذیرش: ۱۳۹۸/۰۷/۳۰.

۱. کارشناس ارشد مهندسی کامپیوتر، رشته کامپیوتر، دانشگاه آزاد واحد زاهدشهر، شهر زاهد شهر، کشور ایران (نویسنده مسئول)

raziyehghasemi478@gmail.com

f_famoori1385@yahoo.com

۲. مربی، رشته کامپیوتر، دانشگاه آزاد واحد زاهدشهر، شهر زاهد شهر، کشور ایران.

مقدمه

رایانش ابری امکان محاسبه، ارتباطات و منابع ذخیره‌سازی نامحدود را به صورت مجازی فراهم می‌کند؛ به طوری که این منابع بر اساس تقاضای مشتریان با پیروی از مدل تجاری "پرداخت در ازای استفاده" تأمین می‌شوند. به منظور رویارویی با افزایش تقاضاهای مصرف‌کننده، ارائه‌دهندگان ابر در حال استقرار مراکز داده بزرگ مقیاس شامل صدها هزار گره در سراسر دنیا هستند. مراکز داده بزرگ امروزی بستری برای میزبانی طیف وسیعی از برنامه‌های کاربردی ترکیبی هستند. در مدل محاسبات ابری استاندارد، کاربر از منابع و سرویس‌های یک مرکز داده استفاده می‌کند. به دلیل تنوع ابرها، نمونه‌های ماشین مجازی و همچنین متفاوت بودن کارایی و قیمت آن‌ها، پیدا کردن ماشین مجازی مناسب برای کاربران یک مسئله پیچیده است. بنابراین به کارگیری مکانیزم کارگزاری منابع در رایانش ابری جهت تطبیق نیازهای کاربرد با منابع ابرها (نمونه‌های ماشین مجازی) به صورت شفاف از دیدگاه کاربر ضروری به نظر می‌رسد (کامل طباخ فریضنی، هاشمی، ۱۳۹۵).

تلاش برای دستیابی به کیفیت سرویس و صرفه‌جویی در مصرف انرژی یکی از مهم‌ترین موضوعاتی است که در مسئله زمان‌بندی منابع مورد توجه قرار گرفته است. تحقق کیفیت سرویس در سطوح مختلف از سطح کسب و کار، طبقه‌بندی و راضی نگه‌داشتن مشتریان و همچنین تحقق کیفیت سرویس در سطح جایگذاری ماشین‌های مجازی و پرداخت جریمه در صورت محقق نشدن مهلت زمانی در تخصیص منابع، مورد توجه می‌باشد و اهمیت این مسئله به خصوص با بالا رفتن استفاده از اینترنت موبایلی و افزایش کاربران موبایل افزایش یافته است. رایانش ابری موبایل به سه لایه تجزیه می‌شود: ابرهای عمومی، ابرچه‌ها و دستگاه‌های تک‌کاره مانند دستگاه‌های بلوتوث. فاصله بیشتر بین کاربر و ابرچه، انرژی و زمان بیشتری برای تخلیه بار مصرف می‌کند. سیگنال‌های رادیویی با پخش شدن در محیط به طور نمایی تضعیف می‌شوند. فاصله بیشتر بین دو جفت به معنای نرخ بیت مؤثر کمتر و کاهش نرخ انتقال است، در نتیجه زمان ارتباط افزایش می‌یابد. افزایش زمان ارتباط منجر به مصرف انرژی بیشتر می‌شود، بنابراین بهره‌وری ابرچه و طول

صف انتظار باید در فرایند تصمیم گیری برای انتخاب ابرچه در نظر گرفته شود. این تصمیم گیری بدین منظور انجام می شود که میانگین زمان خاتمه تمام وظایف و مصرف برق باتری را کم کند (اسمعیلی، میرزایی، ۱۳۹۴).

سه سناریو ممکن در شبکه موبایلی عبارتند از: ۱- تخلیه بار به یک ابرچه: که نزدیک ترین ابرچه با میزان بار کمتر یک انتخاب ایده آل برای کاربر است. هنگامی که تعداد وظایف در انتظار برای اجرا در نزدیکی ابرچه افزایش یابد، به دلیل محدودیت منابع، کاربران تأخیر بیشتری را تجربه می کنند. به هر حال این سناریو همیشه بهترین انتخاب نیست. ۲- تخلیه بار به یک ابر عمومی از طریق ابرچه‌ها: وقتی که تعداد وظایف بالا باشد، همه سرورها در ابرچه‌ها مشغول می شوند و صفوف طولانی از وظایف منتظر اجرا در ابرچه‌ها ایجاد می شود. ۳- تخلیه بار روی یک ابر عمومی از طریق اتصال 3G: که اگر تعداد درخواست‌های اضافه شده زیاد باشد، پهنای باند شلوغ می شود و از دست رفتن بسته‌ها و دوره عکس العمل زیاد می شود. با استفاده از الگوریتم ترکیبی ابتکاری مبتنی بر صف، می توان بهترین مکان برای تخلیه بار وظایف موبایل را با در نظر گرفتن منابع در ابرچه‌ها و مدت ارتباط بین کاربران و ابرچه‌ها، پیدا کرد. مدت ارتباط، به فاصله کاربر از ابرچه وابسته است. کاربر باید حداکثر زمان انتظار قابل تحمل را به عنوان یک پارامتر کیفیت مشخص کند (بن و سوندریا، ۲۰۱۶). هدف اصلی الگوریتم پیشنهادی، بهبود کارایی کلی سیستم با استفاده از کم کردن زمان خاتمه و مصرف برق باتری موبایل است. همچنین تلاش می کند تا تعداد درخواست‌های رد شده، بار ابرچه‌ها را موازنه کند. محاسبات ابری در دنیای امروز به خصوص در عرصه شبکه‌های موبایل، نقش مهمی را بازی می کند و سرویس‌های متعددی را از طریق شبکه فراهم می نماید. ارائه دهندگان سرویس‌های ابری اغلب مراکز داده بزرگی را اداره می کنند. مراکز داده امروزی روز به روز در حال بزرگ تر و پیچیده تر شدن هستند. برنامه‌های کاربردی که توسط این مراکز میزبانی می شوند نیز در حال پیچیده تر شدن بوده و هر یک طیف وسیع و متنوعی از نیازهای ارتباطی را دارا می باشند (هانگ و دوآچا، ۲۰۱۲). ادامه این مقاله به این صورت سازماندهی خواهد شد که در

قسمت اول پژوهشهایی که در گذشته در این حوزه پرداخته شده‌اند، بررسی خواهند شد و در ادامه روش پیشنهادی بیان خواهد شد، در ادامه روش پیشنهادی پیاده‌سازی شده و نتایج آن بیان می‌شود و در نهایت به بحث و نتیجه‌گیری پرداخته خواهد شد.

پیشینه پژوهش

با توجه به رشد روزافزون و سریع سیستم‌های محاسبات ابری موبایل، ارائه الگوریتم‌های جدید و کارآمد برای حل مسئله زمان‌بندی^۱ فرآیندها در این سیستم‌ها ضروری به نظر می‌رسد. گرید یک محیط عملیاتی شامل انواع پردازنده‌ها با قابلیت‌های پردازشی گوناگون است که عملیات زمان‌بندی کارها و تخصیص منابع در آن را با دشواری‌هایی روبرو می‌کند. مدیریت منابع در محیط محاسبات ابری موبایل به علت توزیع جغرافیایی، ناهمگونی و پویا بودن منابع، مسئله پیچیده‌ای است. یافتن بهترین پردازنده برای انجام هر فرآیند از میان تمام پردازنده‌های موجود با قابلیت‌های گوناگون کار پیچیده‌ای است، به طوری که بتوان طبق درخواست کاربر منبعی را پیدا کرد که بیشترین تطابق را با نیاز مطرح شده دارا باشد. در مسئله زمان‌بندی در محیط محاسبات ابری موبایل، هدف یافتن یک روش مناسب برای انتساب فرآیندها به پردازنده‌های ناهمگون است (اپرسکو و کیلمان، ۲۰۱۰).

الگوریتم‌های زمان‌بندی در سیستم‌های کامپیوتری توزیع‌شده و موازی سنتی که معمولاً روی منابع اختصاصی و ناهمگن همچون محاسبات ابری موبایل اجرا می‌شوند، نمی‌توانند در شرایط و موقعیت‌های جدید به خوبی کار کنند. زمان‌بندی وظایف در محاسبات ابری موبایل بسیار پیچیده‌تر از زمان‌بندی منبع محلی است. چون باید منابع در مقیاس بزرگ‌تر را مدیریت و کنترل کند. روش‌های قدیمی زمان‌بندی برای مدیریت منابع در محاسبات ابری موبایل مناسب نیستند. هدف روش‌های قدیمی بالا بردن کارایی سیستم، افزایش بهره‌وری و برآورده کردن نیازهای منابع بود اما در روش‌های جدید

^۱ Schedule

کیفیت سرویس برای کاربران مهم است. هدف زمان‌بندی، تعادل بین هزینه و زمان اجرای کارها روی منابع مختلف می‌باشد (دن بوسچه و مانمچلن، ۲۰۱۳).

زمان‌بندی وظایف و تخصیص منابع در مراکز داده‌ای محاسبات ابری موبایل شامل سه مرحله اصلی است: کشف منبع^۱ که لیستی از منابع ایجاد کرده و اطلاعاتی درباره آن منابع جمع‌آوری می‌کند. انتخاب یک مجموعه خوب از آن‌ها^۲، اجرای کار^۳ که شامل مرحله‌بندی و پاک‌سازی فایل‌های اضافی است. یکی از مهم‌ترین چالش‌ها در زمان‌بندی وظایف ابری موبایل، توازن بار است. هدف از توازن بار، یافتن نگاشتی مناسب از کارها روی پردازنده‌های موجود در سیستم است به طوری که در هر پردازنده مقدار تقریباً مساوی از کارها اجرا شود تا زمان اجرای کلی به کم‌ترین مقدار خود برسد. از آنجا که مسئله توازن بار در کارایی سیستم‌ها بسیار تأثیرگذار است، توجه بسیاری از محققین را در سال‌های اخیر به خود معطوف کرده و کارهای زیادی در این زمینه انجام گرفته است و تلاش‌ها همچنان برای افزایش سرعت سیستم‌ها در انجام محاسبات و بهره‌گیری هرچه بیشتر از منابع در دسترس ادامه دارد. با داشتن شبکه‌هایی با سرعت بالا، تقاضاها برای به دست آوردن کارایی بالای محاسباتی در حال افزایش است (ژانگ، پنگ، لی و لی، ۲۰۱۱).

توازن بار به منزله حذف وظایف از ماشین‌های مجازی زیاد بارگذاری شده و تخصیص آن‌ها به ماشین‌های مجازی کم بارگذاری شده است. زمان‌بندی وظایف می‌تواند روی عملکرد کلی اجرای برنامه‌های یک سیستم تأثیر بگذارد. در الگوریتم‌های زمان‌بندی وظایف ایستا، زمانی که منابع مورد نیاز برآورد می‌شوند، تصمیمات مربوط به زمان‌بندی وظایف در زمان کامپایل ساخته خواهد شد. مزیت استفاده از این الگوریتم سادگی از نظر پیاده‌سازی و سربار است چون هیچ نیازی به نظارت مداوم آمار عملکرد گره‌ها وجود ندارد. الگوریتم‌های ایستا تنها زمانی که تنوع بار در ماشین‌های مجازی کم است به درستی کار می‌کنند. بنابراین، این الگوریتم‌ها برای محیط‌های محاسبات ابری موبایل که در آن تنوع بار در زمان‌های مختلف تغییر می‌کند مناسب نیست. الگوریتم زمان‌بندی وظایف پویا

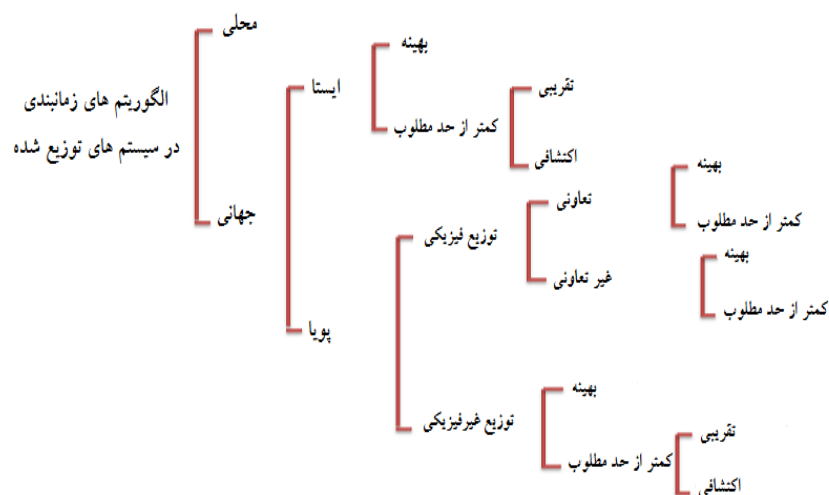
¹ Resource Discovery

² Resource Selection

³ Job Execution

برای توزیع بار کاری در میان گره‌ها در زمان اجرا تغییر ایجاد می‌کنند؛ آن‌ها اطلاعات فعلی بار را در زمان تصمیم‌گیری توزیع استفاده می‌کنند (گارگ، بایا، آناندانیوام و یوو، ۲۰۱۳).

به دلیل تنوع روش‌های ارائه شده برای مسئله زمان‌بندی، مقایسه کمی و کیفی این روش‌ها بسیار مشکل است. قبل از بررسی یا ارائه یک الگوریتم زمان‌بندی، بهتر است جایگاه آن در مقایسه با سایر الگوریتم‌ها مشخص شود. با توجه به این که سیستم توزیع شده نوع خاصی از سیستم‌ها است، الگوریتم‌های زمان‌بندی در گرید در زیرمجموعه‌ای از این طبقه‌بندی (شکل ۱) قرار می‌گیرند (چانک، چانک و لین، ۲۰۰۸).



شکل ۱. طبقه‌بندی الگوریتم‌های زمان‌بندی در گرید

در بالاترین سطح، الگوریتم‌های زمان‌بندی به دو دسته سراسری^۱ و محلی^۲ تقسیم‌بندی می‌شوند. در روش زمان‌بندی محلی در مورد نحوه تخصیص فرآیندها به یک پردازنده و اجرای آن تصمیم‌گیری می‌شود. در روش سراسری با استفاده از اطلاعات سیستم‌ها تخصیص فرآیندها به چندین پردازنده برای بهینه‌سازی کارایی کلی سیستم

^۱. Global
^۲. Local

صورت می‌گیرد. زمان‌بندی توزیع شده در شاخه زمان‌بندی سراسری قرار می‌گیرد (گرین، ۲۰۱۰).

در سطح بعدی در سلسله‌مراتب زمان‌بندی، زمان‌بندی ایستا و پویا است. در الگوریتم‌های ایستا اطلاعات لازم درباره همه منابع موجود در محیط محاسباتی ناهمگن در لحظه زمان‌بندی موجود است. از مزایای این روش سادگی برنامه‌نویسی از دیدگاه زمان‌بند است. در مقابل زمان‌بندی پویا ایده اصلی تخصیص درخواست در زمان اجرای برنامه‌ها است. این روش در مواقعی مورد استفاده قرار می‌گیرد که تخمین هزینه برنامه‌ها یا درخواست‌های وارد شده به سیستم به صورت پویا مشکل است (گرین، ۲۰۱۰).

زمان‌بندی درخواست‌ها به صورت پویا شامل دو قسمت اصلی است: تخمین حالت سیستم (علاوه بر روش تخمین هزینه در روش ایستا) و تصمیم‌گیری. تخمین حالت سیستم شامل جمع‌آوری اطلاعات از سراسر سیستم محاسباتی ناهمگن و سپس تخمین زدن است. بر اساس تخمین به عمل آمده، تصمیم‌انساب یک درخواست به منبع انتخاب انجام می‌شود (گرین، ۲۰۱۰). هر دو نوع زمان‌بندی ایستا و پویا به طور گسترده در محاسبات توزیع شده پذیرفته می‌شوند.

الگوریتم‌های زمان‌بندی از دیدگاه دیگر به دو دسته بلادرنگ (آنلاین) و دسته‌ای تقسیم‌بندی می‌شوند. در حالت بلادرنگ به محض ورود کار، عمل تخصیص کار به یک پردازنده در دسترس توسط زمان‌بند انجام می‌پذیرد در صورتی که در حالت دسته‌ای عمل تخصیص کار طی زمان مشخصی که رخداد زمان‌بندی نامیده می‌شود، انجام می‌شود (گرین، ۲۰۱۰).

دسته دیگر به دو دسته زمان‌بندی بهینه و تقریباً بهینه تقسیم‌بندی می‌شوند. در زمان‌بندی بهینه اختصاص بهینه می‌تواند بر اساس برخی توابع معیار مانند Makespan و استفاده بیشینه از منابع باشد. با توجه به پیچیدگی‌های موجود در محیط‌های محاسباتی ناهمگن بایستی بهینگی الگوریتم ثابت شود (گرین، ۲۰۱۰).

دسته‌بندی دیگر در مورد الگوریتم‌های توزیع یافته در مقابل الگوریتم‌های متمرکز است. در سناریوهای زمان‌بندی پویا، مسئولیت اتخاذ تصمیمات زمان‌بندی سراسری می‌تواند به عهده زمان‌بند متمرکز باشد یا به وسیله زمان‌بندهای توزیعی چندگانه به اشتراک گذاشته شود. در محاسبات توزیع شده ممکن است برنامه‌های کاربردی زیادی ارائه شوند که لازم است به طور همزمان دوباره زمان‌بندی شوند. روش متمرکز، مزیت سهولت اجرا را دارد اما مشکل نبود مقیاس‌پذیری، عدم تحمل خطا، امکان تبدیل شدن به گلوگاه را دارد (قارونی فرد، مرواریدی، دلداری و درباری، ۲۰۱۴).

دسته‌بندی دیگر، الگوریتم‌های تقریبی در مقابل الگوریتم‌های اکتشافی هستند. الگوریتم‌های تقریبی از مدل‌های محاسباتی رسمی استفاده می‌کنند اما به جای جستجوی کل فضای راه‌حل برای یک راه‌حل بهینه، به یک راه‌حل به‌اندازه کافی خوب راضی هستند. در این مورد، در شرایطی که یک معیار برای ارزیابی یک راه‌حل در دسترس باشد، این روش می‌تواند تا موقعی که زمان لازم برای یافتن یک زمان‌بندی مورد قبول کاهش یابد، به کار برده شود (قارونی فرد، مرواریدی، دلداری و درباری، ۲۰۱۴).

در نهایت، دسته‌بندی آخر مربوط به الگوریتم‌های زمان‌بندی تعاونی و غیر تعاونی است. هرگاه یک الگوریتم به عنوان الگوریتم زمان‌بندی توزیعی پذیرفته می‌شود، موضوع بعدی که باید در نظر گرفته شود این است که آیا گره‌هایی درگیر زمان‌بندی کار می‌باشند که به طور تعاونی کار می‌کنند یا به صورت مستقل در حالت کار غیر تعاونی. زمان‌بندهای اختصاصی تنها مثل هویت‌های خودمختار عمل می‌کنند و در تصمیمات با در نظرگیری موضوعات بهینه‌شان مستقل از تأثیرات تصمیم روی بقیه سیستم عمل می‌کنند.

روش‌های زیادی برای انتخاب بهترین داده میزبان و محاسبه میزبان وجود دارد. به عنوان مثال، می‌توان با توجه به نزدیک بودن داده میزبان از نظر فاصله شبکه‌ای با محاسبه میزبان یا با تلاش برای به حداکثر رساندن همکاری ارتباط بین دو داده میزبان، در مورد چگونگی اولویت دسترسی به برخی از مجموعه فایل‌های مشابه تصمیم‌گیری کرد (دشپانده، چان، گوپالان و بیلا، ۲۰۱۴). معیارهای زیادی برای طبقه‌بندی الگوریتم

زمان‌بندی جریان کار وجود دارد که از جمله این معیارها مدل زمان‌بندی، ماهیت الگوریتم زمان‌بندی، نوع الگوریتم زمان‌بندی، هدف بهینه‌سازی و محیط کاربردی است. بر اساس تعداد اهداف بهینه‌سازی، مسئله زمان‌بندی جریان کار می‌تواند به عنوان مدل تک‌هدفی و چندهدفی حل شود. به عنوان مثال، الگوریتم اصلاح هوشمند قطرات آب مدل تک‌هدفی است، الگوریتم ژنتیک چندهدفی مدل چندهدفی برای کشف فضای جستجو و تولید مجموعه راه‌حل پارتو برای انتخاب است (چانگ و رامچاندرا، ۲۰۱۶). جریان کار با بهینه‌سازی تابع‌های مختلف هدفمند تکمیل می‌شود، به طوری که سناریوهای مختلف کاربردی نیازمند الگوریتم متفاوت زمان‌بندی است. بر اساس اهداف بهینه‌سازی الگوریتم، می‌توان آن‌ها را برحسب عملکرد محدود الگوریتم زمان‌بندی، معیارهای کیفیت سرویس کاربر یا ترکیبی از هر دو طبقه‌بندی کرد (گالوی، گابریل و وربسکی، ۲۰۱۶). عملکرد محدود الگوریتم زمان‌بندی، تمرکز بر بهره‌وری انرژی، حفظ تعادل بار و در دسترس بودن و غیره دارد. معیار کیفیت سرویس کاربر محدود معیارهایی هستند که الگوریتم‌های زمان‌بندی را از دیدگاه کاربر و از لحاظ هزینه، زمان و قابلیت اطمینان بهینه می‌کنند؛ مانند الگوریتم ژنتیک مبتنی بر کیفیت سرویس و آگاهی از ترکیب خدمات و الگوریتم‌های بهینه زمان‌بندی ابرهای هیبرید.

الگوریتم‌های زمان‌بندی می‌تواند از نظر نوع الگوریتم، به اکتشافی و فرااکتشافی تقسیم شوند. به عنوان مثال، الگوریتم اکتشافی چندهدفی و الگوریتم زمان‌بندی با مهلت محدود جزء الگوریتم‌های اکتشافی است. برای الگوریتم‌های فرااکتشافی می‌توان الگوریتم اکتشافی مبتنی بر کیفیت سرویس و آگاهی از ترکیب خدمات را نام برد (چن و ژانگ، ۲۰۱۱). هدف از الگوریتم‌های اکتشافی، ارائه راه‌حل در چارچوب یک زمان قابل قبول است که برای حل مسئله مناسب باشد، ممکن است الگوریتم اکتشافی، بهترین راه‌حل واقعی برای حل مسئله نبوده ولی می‌تواند راه‌حل نزدیک به بهترین باشد. الگوریتم‌های اکتشافی می‌توانند با الگوریتم‌های بهینه‌سازی برای اصلاح کارایی الگوریتم ترکیب شوند. الگوریتم فرااکتشافی ترکیبی است از الگوریتم‌های اکتشافی که برای پیدا

کردن، تولید یا انتخاب هر اکتشاف در هر مرحله طراحی می‌شود و راه‌حل خوبی برای مسائلی که مشکل بهینه‌سازی دارند ارائه دهد. الگوریتم‌های فرااکتشافی برخی از فرضیات مسائل بهینه‌سازی را که باید حل شود در نظر می‌گیرند ولی زمان طولانی لازم دارند (لین، جاین و هسو، ۲۰۱۴).

الگوریتم اولین زمان انجام ناهمگن سه فاز دارد: (۱) فاز وزنی - تنظیم وزن‌ها در گره‌های و لبه‌های برنامه (۲) رده‌بندی فاز - ایجاد فهرست در دسته‌های سازماندهی شده (۳) فاز ترسیم کارها را در منابع تنظیم می‌کند. در مرحله وزنی، وزن‌ها در گروه لبه‌ها تنظیم می‌شود. وزن‌ها در گروه‌هایی تنظیم می‌شود که طبق زمان اجرای پیش‌بینی شده کارها، محاسبه می‌شوند و وزن‌ها در لبه‌هایی تنظیم می‌شود که طبق زمان پیش‌بینی تنظیم شود که در آن داده منابع جابجا شده است (بلاویستا، مونتاناری، داس، ۲۰۱۳).

در فاز رده‌بندی، پیمایش رو به بالا انجام می‌شود و مقدار رده هر یک از کارها را تنظیم می‌کند. مقدار رده مساوی وزن گره به علاوه زمان اجرای جانشین‌ها است. زمان اجرای جانشین برای هر لبه با جانشین آن گره برآورد می‌شود و وزن آن را در مقدار رده گره جانشین می‌افزاید و ماکسیمم خلاصه را انتخاب می‌کند. در فاز ترسیم، کارهای متوالی لیست رده‌بندی در منابع ترسیم می‌شود. برای هر کار، منابع، اولین زمان مدنظر را برای تکمیل اجرا انتخاب می‌کنند. در کل این الگوریتم در ابتدا میانگین زمان اجرای هر کار و میانگین زمان ارتباط منابع را با دو کار موفق محاسبه می‌کند. سپس گردش کار برحسب انتخاب منبع، کارها را به ترتیب اولویت زمان‌بندی می‌کند و هر کار در منبعی تنظیم می‌شود که کار را در اولین زمان تکمیل می‌کند.

الگوریتم جدید زمان‌بندی، ترکیبی از هزینه و زمان است و یک مدل چندهدفی که در محیط ابرهیرید استفاده می‌شود. این الگوریتم زمان اجرا می‌شود و هزینه را با ورودی کاربر در نظر می‌گیرد و روی حداقل‌رسانی هزینه‌ای متمرکز است که طبق آن کار موعده مقرر را تعریف کند. عملکرد این الگوریتم به روش Min-Min آنالیز می‌شود (گالوی، گابریل و وربسکی، ۲۰۱۶).

الگوریتم ژنتیک روشی است که خدماتی با بهترین مقادیر کیفیت سرویس ارائه می‌دهد. الگوریتم ژنتیک چهار مرحله: انتخاب، عبور، جهش و بررسی دارد و این الگوریتم جزء الگوریتم‌های چندهدفی می‌باشد. این الگوریتم هزینه جریان کار و معیارهای ساخت را در محیط محاسبات ابری موبایل به حداقل می‌رساند. همچنین محدودیت هزینه‌ای دارد و بیشتر محدودیت کیفیت سرویس در الگوریتم زمان‌بندی در محدودیت هزینه است (چن و ژانگ، ۲۰۱۱).

الگوریتم زمان‌بندی هزینه بهینه ابری^۱ برای ارتباط با زمان‌بندی ابرهای هیبریدی به کار می‌رود. این الگوریتم تصمیم می‌گیرد که منابع باید از ابر عمومی سست شود و به ابر خصوصی برای ارائه توان کافی پردازش و اجرای کار در زمان اجرا داده شده پیوندد. محققین آزمایشاتی انجام دادند که نشان می‌دهد الگوریتم زمان‌بندی هزینه بهینه ابری هزینه را کاهش می‌دهد در حالی که زمان اجرای مدنظر را کسب می‌کند (لین، جاین و هسو، ۲۰۱۴).

یکی از جنبه‌های بسیار مهم برای سیستم ابری، مدل تجارت بازارمحور است که چالش‌های زیادی برای استراتژی‌های زمان‌بندی برنامه کار به همراه دارد. استراتژی زمان‌بندی سلسله‌ای بازارمحور برای سیستم برنامه ابری چندهدفی پیشنهاد شد. در این زمان‌بندی سطح خدمات با تنظیمات کار با سرویس، و زمان‌بندی سطح کار با بهینه‌سازی تنظیم کار با ماشین‌های مجازی مراکز ابری سروکار دارد (لیو، شاو، جینگ و کیو، ۲۰۱۵). الگوریتم انشعاب جریان کار تطبیقی^۲ (AWS) به این صورت عمل می‌کند که زمانی که زمان‌بندی کار را دریافت می‌کند، ابتدا آن DAGG کار را به منطق منتقل می‌کند تا CCR آن و تعداد دسته‌های منبع را محاسبه کند، که در آن CCR میانگین نسبی هزینه محاسبه در هزینه ارتباط است. مقدار بالای CCR نمودار کار محاسبه جامع است. پس، زمان‌بندی دسته‌های منبع N را انتخاب می‌کند که بالاترین رده N را مطابق شاخص دارد. تکرار درصد رشد نموداری هر گره را در G چک می‌کند تا بررسی کند آیا گره می‌تواند در زیر نمودار G قرار گیرد که با بررسی بزرگ‌ترین لبه ارتباطی شروع می‌شود

¹ Hybrid Cloud Optimized Cost (HCOC)

² Adaptive Workflow Splitting

که دو گره آن چک نشده است. اگر هیچ لبه‌ای نباشد به این معنی است که کل گره‌های چک نشده در G همدیگر ایزوله شده است، این کارها در صورت ممکن در زیر نمودار ظاهر می‌شود.

هدف از پیمایش نمودار، بروزرسانی بیشتر کارها در زیر نمودار فعلی است و محدودیت آستانه منبع را مختل نمی‌کند و در عین حال، کارهای غیرضروری را دنبال می‌کند و نمی‌تواند توازن را افزایش دهد ولی هزینه ارتباط درون دسته را افزایش می‌دهد. (ژانگ، رن و شو، ۲۰۱۶). برای دست کاری گره کار در پیمایش، از پشته استفاده می‌شود. اگر این لبه نمی‌تواند در G محدود آشکار شود، گره مسئول مشخص می‌شود به این معنی که زیرنمودار جدید درست می‌شود. زمان بندی حلقه گره‌ها را در G تکرار می‌کند و در دسته‌های منبع تنظیم می‌کند و الگوریتم پایان می‌یابد، زمان بندی هر زیرنمودار را در دسته منبع تنظیم شده‌اش گزارش می‌کند.

کارهای قبلی که در زمینه‌ی زمان بندی ماشین‌های مجازی و پارتیشن بندی کد وجود دارند دارای اهداف مختلفی هستند. برخی از الگوریتم‌ها درصدد انتخاب میزبان مناسب برای درخواست ماشین‌های مجازی با هدف کاهش مصرف انرژی در مراکز داده می‌باشند. برخی از کارهای قبلی از راه‌حل‌های بهینه‌سازی چندمنظوره برای حل مسئله زمان بندی وظایف و پارتیشن بندی کد استفاده کرده‌اند. در برخی دیگر از کارهای تکمیل شده، هدف کاهش هزینه برای کاربران درخواست دهنده ماشین مجازی است. در این روش‌ها فراهم کننده سرویس تعدادی ماشین مجازی را از ابتدا برای خود رزرو می‌کند و در صورت نیاز ماشین اضافی درخواست می‌دهد. تابع هدف برای زمان بندی ماشین مجازی و پارتیشن بندی کد در این روش بر مبنای کاهش هزینه است. همچنین برخی از کارها با هدف کاهش تعداد ماشین‌های فیزیکی فعال سعی بر کاهش مصرف انرژی و هزینه دارند و این گونه الگوریتم‌ها با افزایش تعداد درخواست‌ها کارایی چندانی ندارند. در سال‌های اخیر برخی از الگوریتم‌ها با هدف کاهش ترافیک موجود روی ماشین‌های فیزیکی روش پویایی را برای زمان بندی ماشین‌های مجازی ارائه می‌دهند و این روش‌ها با کمترین نیاز به

مهاجرت، روش بهینه‌ای برای زمان‌بندی وظایف و پارتیشن‌بندی کد ماشین‌های مجازی می‌باشند (اسمعیلی، میرزایی، ۱۳۹۴).

اغلب روش‌های قبلی که برای جایابی ماشین مجازی بیان شده است با هدف تأمین و بهبود کارایی یکی از معیارهای کیفیت سرویس می‌باشد. در این پژوهش روشی را ارائه می‌دهیم که مسئله زمان‌بندی وظایف و پارتیشن‌بندی کد را با هدف افزایش توازن بار که یکی از مهم‌ترین معیارهای کیفیت سرویس است در نظر می‌گیرد. الگوریتم پیشنهادی به ما کمک می‌کند تا با استفاده حداکثری از منابع، سبب توازن مناسب بار در مراکز داده‌ای ابری سیار شویم و در عین حال، با در نظر گرفتن قابلیت‌های هر یک از ماشین‌های فیزیکی و استفاده از روش مناسب جهت توزیع درخواست‌های ماشین علاوه بر استفاده بهینه از منابع سبب افزایش توازن بار و تأمین معیارهای کیفیت سرویس شویم.

روش‌شناسی پژوهش

تحقیقات بسیاری وجود دارند که به طور کامل در حوزه الگوریتم‌های زمان‌بندی انجام شده‌اند. اکثر آن‌ها بر تعادل بار برای سرورهای وب سنتی تمرکز می‌کنند. یکی از مشکلات چالش‌برانگیز برنامه‌ریزی در مراکز داده ابری در نظر گرفتن هر دو عملیات تخصیص و مهاجرت (انتقال) ماشین‌های مجازی با قابلیت پیکربندی مجدد و ویژگی‌های یکپارچه‌ی ماشین‌های فیزیکی در حال میزبانی است. بر خلاف الگوریتم‌های برنامه‌ریزی برقراری توازن بار سنتی که تنها سرورهای فیزیکی با در نظر گرفتن معیارهای کارایی متفاوت و با مدنظر قرار دادن عواملی چون بار پردازنده، حافظه و پهنای باند شبکه یکپارچه برای هر دو ماشین فیزیکی و ماشین‌های مجازی و... در این پژوهش معیار یکپارچه‌ای برای برقراری توازن و تعادل بار بر روی هر سرور ایجاد می‌شود. در این پژوهش یک الگوریتم زمان‌بندی یکپارچه و پویای منابع بر اساس الگوریتم تکاملی بهینه‌سازی ازدحام ذرات معرفی خواهد شد که پردازنده، حافظه و پهنای باند یکپارچه برای هر دو ماشین فیزیکی و ماشین‌های مجازی را اصلاح می‌کند.

در محیط ابری، معماری تخصیص وظیفه یک مدل سه لایه است، یعنی درخواست وظیفه و کار، مدیریت منابع و اجرای وظیفه. در لایه اول، کاربران با سیستم تعامل می کنند؛ در لایه میانی، مدیریت منبع یک نقش قابل توجه و برجسته در تقسیم کار به وظایف، تخصیص وظایف و مدیریت منابع بازی می کند؛ در آخرین لایه وظایف به عنوان توالی های از پیش تعریف شده اجرا می شوند. مدیریت مراکز داده ای کلیدی برای تخصیص وظایف است.

هدف اصلی جایابی، به حداکثر رساندن استفاده از منابع، به حداقل رسانی زمان اجرای ماشین های مجازی و با افزایش توازن بار در محیط محاسباتی ابری پس از رفع محدودیت های ماشین های مجازی و منابع سیار می باشد. ما به دنبال برقراری تعادلی مناسب بین سطح مورد نیاز کیفیت خدمات می باشیم. در این پژوهش، یک روش زمان بندی و پارتیشن بندی کد با حالت پویای مبتنی بر تکامل چند هدفی در شبکه را پیشنهاد داده ایم. رویکرد جایابی ما از روش بهینه سازی چند هدفی با استفاده از الگوریتم بهینه سازی ازدحام ذرات برای پیدا کردن جایابی های کارآمد استفاده کرده است. این روش دست به کاوش فضای جستجو به طور روشن زده تا از ایستایی جلوگیری کرده و راه حل نزدیک به بهینه را تولید نماید.

پیدا کردن راه حل نزدیک به بهینه برای مشکل زمان بندی وظایف و پارتیشن بندی کد منعطف با بیش از یک هدف در یک روش کارآمد از نظر زمانی، کاری دشوار است. محیط ابر دارای ماهیت پویا بوده و تخصیص دوباره ماشین های مجازی در آن کاملاً آشکار کننده این وضعیت می باشد. هدف این پژوهش، ارائه یک جایابی بهتر به مجریان ابر بوده که دارای وضعیت بهتری از نظر توازن بار و کیفیت خدمات خواهد بود. این جایابی کننده می تواند از عهده رفتار پویای منابع، محدودیت های منبع و محدودیت های مربوط به ماشین های مجازی پیشین برآید.

روش جایابی ارائه شده در این پژوهش، مزیت بیشتری در رابطه با اکثر ویژگی های ابر چه از نظر سرویس دهنده و چه از نظر مشتریان فراهم می سازد. محیط پویای شبکه روش

پیشنهادی را به یک روش جایابی پویا با زمان واقعی تبدیل می کند. یک روش گروه بندی ماشین های مجازی برای گروه بندی ماشین های مجازی به خوبی شبکه ای شده به منظور تسهیل در محاسبات پیچیده پیشنهاد شده است. انجام آزمایشات روی مجموعه های مختلف داده ها و پارامترهای گوناگون نشان دهنده کارآمدی معیارهای جایابی چندهدفی و استخراج عملکرد از منبع شبکه بوده است. رویکرد ما در رابطه با حل مشکلات بالا با استفاده از الگوریتم تکاملی برای کوچک کردن حیطه اهداف چندگانه به اندازه ای مناسب بوده که راهبرد جایابی نزدیک به بهینه را به موقع پیدا کند. الگوریتم بهینه سازی ازدحام ذرات به عنوان یک چهارچوب بنیادین برای ماژول زمان بندی وظایف و پارتیشن بندی کد ما به کار گرفته شده است. ثابت شده است که زمان بندی وظایف و پارتیشن بندی کد یک مسئله چندجمله ای غیرقطعی است.

به دلیل افزایش قابلیت محاسباتی پردازشگرها در پنج سال گذشته، روش جایابی مبتنی بر الگوریتم بهینه سازی ازدحام ذرات می تواند به عنوان یک روش جایابی زمان واقعی عمل کند. این جایابی یک روش جستجو بوده که در آن شایستگی به صورتی پویا بر مبنای واریانس های مقادیر تناسب در هر نسل تغییر می کنند. این روش جایابی، به حداقل سازی زمان تکمیل ماشین های مجازی نیز توجه می کند. در این پژوهش بر مبنای الگوریتم تکاملی چندهدفی، مسئله جایابی منبع در شبکه را به مسئله جایابی پروژه محدود از نظر منبع تبدیل کرده ایم. در اینجا ما از سازوکار جایابی پویا، عامل پیشرفته یادگیری استفاده کرده ایم که به دنبال بهینه سازی پنج هدف می باشد. ماژول چندهدفی زمان بندی وظایف و پارتیشن بندی کد با راهبرد گروه بندی ماشین های مجازی مبتنی بر نمودار برای ماشین های مجازی به خوبی دانه دانه شده، این فرآیند را آسان می کند.

به منظور ارزیابی جنبه های مثبت و منفی رابطه نقشه برداری از کارگران M با وظایف $TaskNum$ ، باید کل زمان مصرفی یک مدل نقشه برداری معین کار شود و مقایسه متقابل انجام شود. به منظور آنالیز کل زمان اجرای وظایف ارائه شده، در ابتدا تعداد وظایف در ماشین مجازی t به عنوان $TaskNum(t)$ به طور کمی تعیین می شود که توسط فرمول یک

مشخص می‌شود که J تعداد کل وظایف است، N مخفف شماره ماشین‌های فیزیکی در مراکز داده‌ای ابری است.

$$taskNum(t) = \sum_{t=1}^J TaskNum(t) \quad \text{رابطه (۱)}$$

برای تمایز بین این وظایف، وظیفه j^{th} در وظیفه i^{th} با m با استفاده از رابطه (۲)

شناسایی می‌شود،

$$m = \sum_{k=1}^{i-1} TaskNum(k) + j \quad \text{رابطه (۲)}$$

ماتریس ETC (tasknum × workernum) (زمان مورد انتظار برای محاسبه) با توصیف زمان مصرفی وظیفه i^{th} انجام شده با محاسبه کننده j^{th} را معرفی می‌کند که tasknum به شماره وظایف اشاره دارد و workernum تعداد واحدهای محاسباتی را بیان می‌کند. بنابراین، با توجه به رابطه رمزگشایی شده بین پردازشگرها و وظایف و ماتریس ETC، زمان اجرای کار t^{th} ایجاد می‌شود.

$$JopTime(t) = \max_{i=1}^{TaskNum(t)} \sum_{j=1}^k TaskNum(j, i) + j \quad \text{رابطه (۳)}$$

که k نشان‌دهنده وظیفه i^{th} در موقعیت تخصیص داده شده به ماشین مجازی t است، TaskTime(j,i) نشان‌دهنده هزینه اجرای وظیفه i در ماشین فیزیکی j است. میانگین زمان اجرا یک ماشین مجازی توسط رابطه (۴) مشخص می‌شود.

$$augJobTime = \frac{\sum_{t=1}^J JobTime(t)}{J} \quad \text{رابطه (۴)}$$

محدوده و ظرفیت وظیفه شامل سه بخش است، یعنی زمان محاسبه وظیفه، دیسک

I/O زمان و زمان انتقال داده‌ها از راه دور. برای یک وظیفه i، $1 \leq i \leq taskNum(t)$ ، نشان‌دهنده زمان انجام وظیفه i^{th} است و چنین مشخص می‌شود.

$$T_i = T_{computing} + T_{\frac{disk}{O_t}} + T_{transmission} \quad \text{رابطه (۵)}$$

$T_{computing}$ هنگامی که با دیگر هزینه‌های زمان در خوشه‌ها محاسبه شود، نسبتاً کوتاه است. دیسک I/O زمان مورد نیاز برای انتقال داده‌ها از دیسک محلی را نشان می‌دهد که با محیط سخت‌افزار خاصی مشخص می‌شود. $T_{transmission}$ هزینه زمانی انتقال داده‌ها از راه

دور است. در این پژوهش، تأخیر زمانی شبکه محاسبه نمی شود، از ماتریس DTC (taskMum × N) برای نشان دادن هزینه انتقال داده ها استفاده می شود که DTC[i,j] هزینه وظیفه برنامه ریزی شده i^{th} برای ماشین فیزیکی j^{th} است، اگر یک تکرار داده در گره j^{th} وجود داشته باشد، $DTC[i,j]=0$. به طور پیش فرض سه استراتژی تکرار داده در سیستم توزیع شده اتخاذ می شود. به طوری که کل زمان مصرفی وظایف ماشین مجازی به صورت رابطه (۶) است:

$$TotalTime = \max_{w=1}^N \sum_{i=1}^N node(w, i) \quad \text{رابطه (۶)}$$

که گره (w, i) به هزینه زمانی وظیفه i^{th} اجرا شده در ماشین مجازی w^{th} اشاره دارد که شامل محاسبه، دیسک I/O و انتقال داده از راه دور است و n تعداد کل وظایف در این گره است.

میانگین بهره وری پردازنده در یک سرور منفرد i با CPU_i^U نشان داده می شود. برای به دست آوردن این معیار از بهره وری پردازنده در طی دوره مشاهده میانگین گرفته می شود. برای محاسبه برای مثال، اگر دوره مشاهده یک دقیقه و کاربرد پردازنده هر ۱۰ ثانیه ثبت شود، پس CPU_i^U میانگین شش مقدار ثبت شده از سرور i است. CPU_i^n تعداد کل CPU های سرور i باشد، آنگاه میانگین کاربرد همه CPU ها در یک مرکز داده ابری به صورت رابطه ۷ محاسبه می شود:

$$CPU_u^A = \frac{\sum_i^N (CPU_i^U) CPU_i^n}{\sum_i^N CPU_i^n} \quad \text{رابطه (۷)}$$

که N تعداد کل سرورهای فیزیکی در یک مرکز داده ابری است. به طور مشابه، میانگین کاربرد حافظه، پهنای باند شبکه سرور i ، همه حافظه ها و همه پهنای باند شبکه در یک مرکز داده ابری می تواند به ترتیب به صورت MEM_u^A , NET_u^A , MEM_u^U , NET_u^U تعریف شود.

در این تحقیق، الگوریتم بهینه سازی ازدحام ذرات چندهدفه روی مسئله زمان بندی کارهای غیروابسته ارائه شده است. در این تحقیق از یک مدل ذره خاص استفاده شده

است که یک راهبرد زمان‌بندی وظایف و پارتیشن‌بندی کد بر روی مراکز داده‌ای ابری که برآورده کننده محدودیت‌های مختلف اجرایی بوده، توسط ذره ارائه شده است.

الگوریتم‌های بهینه‌سازی ازدحام ذرات کلاسی از الگوریتم‌های جستجوی بهینه هستند که تکامل بیولوژیکی و مکانیسم هوش ازدحامی را شبیه‌سازی می‌کنند. بعد از این که جمعیت اولیه تولید می‌شود، راه‌حل‌های تقریبی بهتر و بهتری بر اساس سازگاری از نسلی به نسل دیگر تکامل می‌یابد. در طی هر نسل، فرد بر اساس سازگاری افراد مختلف در دامنه مشکل خاصی انتخاب می‌شود. بر اساس وضعیت واقعی محاسبات ابری. این پژوهش استراتژی زمان‌بندی و زمان‌بندی وظایف و پارتیشن‌بندی کد از طریق الگوریتم بهینه‌سازی ازدحام ذرات بهبود یافته پیشنهاد می‌کند.

یک طرح شماتیک برنامه‌ریزی می‌تواند توسط ذره‌ای بیان شود که به شکل گره-وظیفه کد می‌شود (شفیع، لطیف، گدافی و مادنی، ۲۰۱۶). طول ذره دقیقاً برابر با تعداد ماشین‌های مجازی است. مقدار یک ژن برابر با تعداد ماشین‌های فیزیکی است که در آن وظیفه این ژن مطابقت است با اجرا. با توجه به ذره زیر:

$$\{1,3,2,4,3,1,4,2,3,1,3,2,4,1\}$$

اولین ماشین مجازی در مرکز داده‌ای ابری اول اجرا می‌شود، دومین ماشین مجازی در مرکز داده‌ای ابری سوم اجرا می‌شود. منظور از $Node_i$ بیانگر ماشین فیزیکی i ام است که بیانگر ماشین‌های مجازی متناسب شده به آن می‌باشد.

$$Node_1 : \{1,6,10\}, Node_2 : \{3,8,12\},$$

$$Node_3 : \{2,5,9,11\}, Node_4 : \{4,7,13\},$$

الگوریتم‌های ازدحام ذرات ساده به صورت تصادفی جمعیت اولیه را ایجاد می‌کنند و افراد عملکرد پایینی دارند. در این پژوهش، الگوریتم حریم‌ناهنه^۱ استفاده می‌شود. جمعیت اولیه از طریق انجام انتخاب بهینه بر اساس وضعیت فعلی ایجاد می‌شود.

توازن بار عبارت از توزیع درخواست‌ها میان چندین ماشین مجازی است که همگی سرویس یکسانی را برای کاربران ارائه می‌دهند. فرایند توازن بار به صورت شفاف انجام

¹ Greedy Algorithm

می گیرد و به این معنی است که فرایند از دید کاربران مخفی صورت می گیرد و کاربران از چگونگی توزیع بار و این که از کدام ماشین مجازی واقع در ابر پاسخ می گیرند هیچ اطلاعی ندارند. در این پژوهش یک تابع سازگاری بر اساس چندین هدف کارایی مختلف تعریف می شود. اهداف کارایی شامل موارد زیر است:

(۱) زمان مصرفی کمتر (۲) واریانس کمتر برای بار مراکز داده ای و ماشین های فیزیکی (۳) زمان پاسخ (۴) میزان حافظه باقیمانده.

- زمان مصرفی کمتر

برای محاسبه معیار کل از جمع زمان اجرای ماشین های مجازی اختصاص داده شده به یک ماشین فیزیکی و مراکز داده ای استفاده می شود که این امر از طریق رابطه (۶) فراهم شده و با $totalTime_{CDC}$ نشان داده می شود.

- واریانس کمتر برای بار مراکز داده ای و ماشین های فیزیکی

برای محاسبه معیار دوم می توان اظهار نمود که حجم بار بر روی ماشین برابر با مجموعه فضای اشغال شده تراکنش های صف ورودی ماشین مجازی می باشد و موجب افزایش مصرف حافظه در ماشین مجازی می شود.

- زمان پاسخ

زمان پاسخ از مهمترین پارامترهای مورد بحث در مورد سیستم ها است و سیستم های محاسبات ابری جدا از این ویژگی نیستند. این زمان عبارت است از زمان تحویل یک تراکنش به یک ماشین مجازی تا زمان اولین اجرای آن. زمان پاسخ مربوط به یک ماشین مجازی بر اساس رابطه ۲ محاسبه می شود که در این رابطه PT_i مدت زمان پردازش یک تراکنش، r_i برابر با زمان دریافت پاسخ و s_i برابر با زمان ارسال تراکنش به ماشین مجازی است.

$$WT = \frac{\sum_{i=1}^m r_i - s_i - PT_i}{m} \quad \text{رابطه ۸}$$

- میزان حافظه باقیمانده

حاکمی از مقدار فضای باقیمانده با وجود تمامی تراکنش‌های موجود در صف تراکنش‌های ماشین مجازی است. این پارامتر به دلیل پیشگیری از مشکل کمبود حافظه دارای اهمیت زیادی است. از این رو، ما این پارامتر را مدنظر قرار دادیم. این معیار با Mem_{ava} نشان داده می‌شود و به معنای میزان حافظه در دسترس است.

روش پیشنهادی باید توازن مناسبی بین این اهداف برقرار سازد. همچنین این اهداف ممکن است بسته به کاربردهای مختلف دارای اولویت‌های متفاوتی باشند. بنابراین برای رسیدن به این هدف، باید به کاربر این اختیار را داد تا توازن مناسبی بین این فاکتورها برقرار سازد و برای این کار کاربر مجموعه‌ای از وزن‌ها را به هر یک از این اهداف می‌دهد. بنابراین تابع شایستگی روش پیشنهادی، در رابطه ۹ نمایش داده شده است:

$$\text{Fitness} = \alpha \times ILB + \beta \times \text{totalTime} + \varphi \times WT + \rho \times Mem_{ava} \quad (\text{رابطه ۹})$$

که در آن α و β ضرایب مختلف وزنی می‌باشد که مربوط به هر یک از اهداف کارایی بوده و در آن $\alpha + \beta + \varphi + \rho = 1$ است.

مراحل اصلی الگوریتم جایابی پیشنهادی به صورت زیر است:

مرحله ۱: به دست آوردن اطلاعات کاربرد دوره مشاهده فعلی، شامل [کاربرد CPU، کاربرد حافظه، کاربرد شبکه].

مرحله ۲: پذیرش درخواست وظیفه جدید (از بالاترین به پایین‌ترین اولویت صف‌ها به صورت صف‌های در حال انتظار، بهینه‌سازی، حذف).

مرحله ۳: بررسی صف در حال انتظار، اگر خالی نباشد، همان صف در حال انتظار، سپس مرحله ۷ و فرایند تخصیص خاص برداشته می‌شوند. اگر عمل تخصیص موفق نشود، پس به مرحله ۸ می‌رود. اگر عمل تخصیص موفق باشد، به مرحله ۱ باز می‌گردد.

مرحله ۴: بررسی صف درخواست، با توجه به زمان شروع وظیفه اگر خالی نباشد، پس تخصیص به دنبال الگوریتم تخصیص برای مرحله ۷ شروع می‌شود، اگر موفق‌آمیز باشد، درخواست در صف حذف قرار می‌گیرد.

مرحله ۵: بررسی صف بهینه سازی، اگر خالی نباشد به مرحله ۹ می رود.
مرحله ۶: بررسی صف حذف، با توجه به زمان اتمام وظیفه آن را حذف می کند.
مرحله ۷: اجرای الگوریتم جایابی بر اساس الگوریتم بهینه سازی ازدحام ذرات با تابع هدف پیشنهادی.

مرحله ۸: اگر همه سرورهای فیزیکی نتوانند تخصیص داده شوند، پس وظیفه به صف انتظار در مد اولین ورود- اولین- سرویس^۱ اضافه می شود.

مرحله ۹: فرایند مهاجرت (انتقال): با توجه به آستانه های کاربرد از پیش تعیین شده CPU، حافظه، پهنای باند، الگوریتم سرور فیزیکی را با بالاترین بار و ماشین مجازی با کمترین بار برای مهاجرت با استفاده از الگوریتم تخصیص را می یابد. مهاجرت تا زمانی که کاربرد سرور فیزیکی زیر آستانه است، یا تکراری از مهاجرت همان ماشین مجازی وجود دارد، ادامه دارد.

مرحله ۱۰: بعد از بررسی همه صف ها، به مرحله ۱ برگرد.

برای پیاده سازی روش پیشنهادی از نرم افزار شبیه ساز محیط محاسبات ابری CloudSim 3.0.3 در محیط زبان برنامه نویسی Java استفاده شده است. این شبیه سازی روی دستگاهی کامپیوتری با پردازنده corei7 با میزان حافظه اصلی ۸ گیگابایت و بر روی سیستم عامل ویندوز ۱۰ اجرا شده است.

برای شبیه سازی روش پیشنهادی از مدل استاندارد SDSC^۲ مربوط به پایگاه داده سن دیگو استفاده شده است. این مجموعه داده متشکل از ۷۳۴۹۶ وظیفه است که مشخصات هر یک از این وظایف از قبل مشخص است. در این پژوهش از ویژگی های زمان ورود، زمان اختصاص پردازنده، زمان اجرای کل، میزان عملیات ورودی، خروجی استفاده می شود و بر مبنای این مشخصات عملیات زمان بندی ایجاد شده و فیلدهایی چون زمان اجرا، زمان پاسخ و زمان اجرای کل تنظیم می شود و جهت ارزیابی روش پیشنهادی مورد استفاده قرار می گیرد. در شبیه سازی های به کار رفته، تعداد وظایف بین ۲۰۰ تا ۲۰۰۰ در

^۱ First-Come-First-Service Fashion

^۲ The San Diego Supercomputer Center (SDSC)

نظر گرفته شده است. تعداد ماشین‌های مجازی برابر با ۵۰ بوده که هر یک دارای یک پردازنده است. تعداد مراکز داده‌ای و میزبان‌ها برابر با ۵۰ در نظر گرفته شده است. برای ارزیابی کارایی روش پیشنهادی از معیار زمان اجرای کل استفاده می‌شود که بیانگر بزرگ‌ترین زمان در بین زمان‌های خاتمه منابع محاسباتی است. در واقع این معیار بیانگر زمانی است که می‌توان از اتمام تمامی وظایف محوله به مراکز داده‌ای اطمینان حاصل کرد.

یافته‌های پژوهش

روش پیشنهادی بر پایه الگوریتم فراابتکاری بهینه‌سازی ازدحام ذرات چندهدفه با برخی از روش‌های موجود در زمینه زمان‌بندی فعالیت‌ها در محیط محاسبات ابری مورد ارزیابی قرار می‌گیرد. برای مقایسه از الگوریتم ارائه شده در (شفیع، لطیف، گدافی و مادنی، ۲۰۱۶)، (کومار و ورما، ۲۰۱۲) و (توفیق، سیسی، کشک و توکی، ۲۰۱۳) استفاده شده است. دلیل استفاده از روش ارائه شده در (شفیع، لطیف، گدافی و مادنی، ۲۰۱۶) این است که از یک الگوریتم تکاملی قهرمانی در لیگ (LCA) بهره می‌برد. برای اولین مقایسه، زمان اجرای کل روش پیشنهادی در مقایسه با روش‌های پیشین بر اساس زمان اجرای کل بر حسب تغییر در تعداد وظایف مقایسه می‌شود. روش ارائه شده در (کومار و ورما، ۲۰۱۲) نیز از الگوریتم ژنتیک و در (توفیق، سیسی، کشک و توکی، ۲۰۱۳) از روشی مبتنی بر الگوریتم بهینه‌سازی ازدحام ذرات استفاده می‌نماید. برای این کار تعداد وظایف همانند روش ارائه شده در (شفیع، لطیف، گدافی و مادنی، ۲۰۱۶)، از ۲۰۰ تا ۲۰۰۰ و با فاصله ۲۰۰ تایی قرار گرفته و در هر حالت زمان اجرای کل مورد محاسبه و ارزیابی قرار گرفته است. نتایج به دست آمده در جدول ۱ نشان داده شده‌اند.

جدول ۱. مقایسه زمان اجرای روش های مختلف

تعداد وظایف	زمان اجرای کل روش پیشنهادی	زمان اجرای کل روش لطیف، گدافی و مادنی، (۲۰۱۶)	زمان اجرای کل روش (کومار و ورما، سیسی، کشک و توکی، (۲۰۱۳)	زمان اجرای کل روش ارائه شده در (توفیق، (۲۰۱۲)
۲۰۰	۳۵	۴۰	۵۱	۴۳
۴۰۰	۸۰	۹۰	۱۰۲	۹۸
۶۰۰	۱۵۰	۱۷۰	۱۹۲	۱۸۵
۸۰۰	۱۹۵	۲۳۰	۲۶۵	۲۵۴
۱۰۰۰	۲۸۵	۳۲۰	۳۸۲	۳۶۱
۱۲۰۰	۳۷۰	۴۴۰	۵۰۳	۴۸۹
۱۴۰۰	۵۴۰	۵۹۰	۶۴۵	۶۲۳
۱۶۰۰	۶۲۰	۶۸۰	۷۵۱	۷۳۰
۱۸۰۰	۷۳۵	۸۰۰	۸۷۸	۸۴۰
۲۰۰۰	۸۱۰	۸۸۰	۹۵۳	۹۲۱

همان طور که از جدول ۱ به خوبی مشخص است، روش پیشنهادی این پژوهش بر اساس مقایسه با روش ارائه شده در مطالعات قبلی، کارایی مناسبی را از خود نشان داده است و در تمامی حالت اجرایی مختلف بر حسب تعداد وظایف متفاوت دارای عملکرد مناسبی بوده و دارای زمان اجرای کمتری نسبت به سایر روش های موجود دارد. این امر به علت استفاده از الگوریتم بهینه سازی ازدحام ذرات چندهدفه با تابع هدف مناسب در این پژوهش است که به خوبی قادر به جستجوی فضای مسئله و اجرای مناسب عمل زمان بندی است. نمودار ارائه شده در شکل ۲ نیز این مقایسه را به خوبی نشان می دهد.



شکل ۲. نمودار مقایسه زمان اجرای روش پیشنهادی و سایر روش‌ها

روش پیشنهادی با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات چندهدفه، بار کاری را با انتقال پویای حجم کار محلی از یک ماشین به ماشینی در گره راه دور یا ماشینی که کمتر مورد استفاده قرار می‌گیرد، موازنه می‌کند و از این طریق زمان اجرای کل را به خوبی کاهش داده است. این عمل رضایت کاربر را حداکثر، زمان پاسخ را حداقل، بهره‌برداری از منابع را افزایش، تعداد رد کارها را کاهش و کارایی سیستم را بالا می‌برد که در این بخش زمان اجرای کل با سایر روش‌ها مورد مقایسه قرار گرفت و در ادامه سایر اهداف کارایی مورد بررسی قرار می‌گیرد. این الگوریتم نه تنها تعادل بار را انجام می‌دهد، بلکه زمان پاسخ وظایف نگاشت شده به ماشین‌های مجازی سربار گذاری شده را نیز در نظر می‌گیرد و زمان بندی و توازن بار را به خوبی انجام می‌دهد و انتظار می‌رود که به خوبی قادر به توزیع متوازن وظایف باشد.

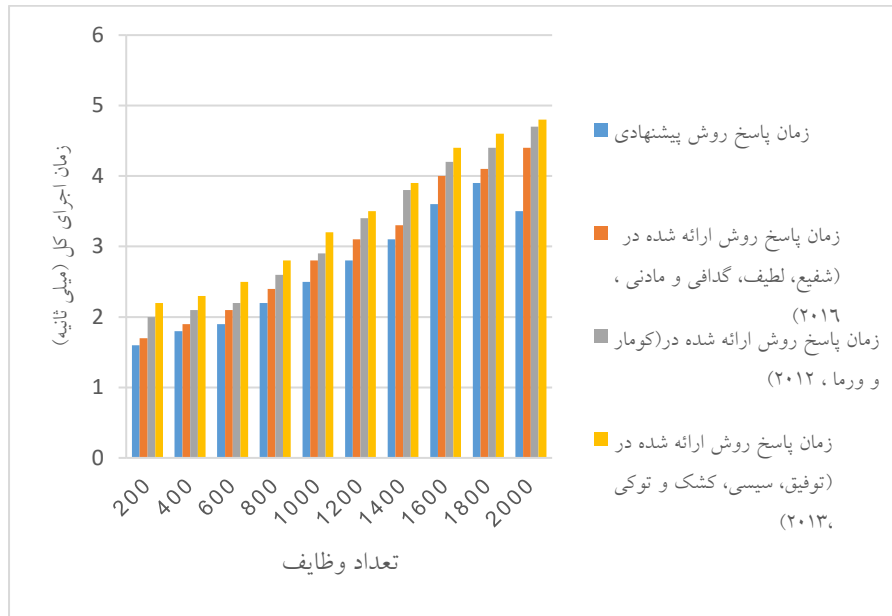
زمان پاسخ بیانگر فاصله زمانی بین درخواست اجرا و شروع اجرا توسط پردازنده‌های موجود در مراکز داده‌ای ابری میزبان است. این معیار به یک معیار استاندارد در مطالعات موجود در حوزه زمان بندی تبدیل شده است. در این بخش مقایسه‌ای بین روش پیشنهادی و مطالعات انجام شده قبلی بر حسب زمان پاسخ ارائه می‌شود. در واقع این معیار بیان‌گر

نسبت عدالت از طریق پاسخگویی در استفاده مناسب از منابع می باشد. برای مقایسه روش پیشنهادی با مطالعه پیشین، حالتی بر حسب تعداد وظایف مختلف در نظر گرفته می شود (جدول ۲) نتایج این مقایسه ها را به خوبی نشان می دهند.

جدول ۲. مقایسه زمان پاسخ روش های مختلف

تعداد وظایف	زمان پاسخ روش پیشنهادی	زمان پاسخ روش ارائه شده در (شفیع، لطیف، گدافی و مادنی، ۲۰۱۶)	زمان پاسخ روش ارائه شده در (کومار و ورما، ۲۰۱۲)	زمان پاسخ ارائه شده در (توفیق، سیسی، کشک و توکی، ۲۰۱۳)
۲۰۰	۶/۱	۷/۱	۰/۲	۲/۲
۴۰۰	۸/۱	۹/۱	۱/۲	۳/۲
۶۰۰	۹/۱	۱/۲	۲/۲	۵/۲
۸۰۰	۲/۲	۴/۲	۶/۲	۸/۲
۱۰۰۰	۵/۲	۸/۲	۹/۲	۲/۳
۱۲۰۰	۸/۲	۱/۳	۴/۳	۵/۳
۱۴۰۰	۱/۳	۳/۳	۸/۳	۹/۳
۱۶۰۰	۶/۳	۰/۴	۲/۴	۴/۴
۱۸۰۰	۹/۳	۱/۴	۴/۴	۶/۴
۲۰۰۰	۵/۳	۴/۴	۷/۴	۸/۴

نتایج نشان می دهد که الگوریتم پیشنهادی در کلیه حالات و بر حسب تعداد وظایف مختلف، زمان پاسخ را نسبت به روش ارائه شده قبلی کاهش می دهد. روش پیشنهادی به خوبی زمان اجرای کل را کاهش داده است و در این قسمت نیز دیگر معیار ارزیابی کارایی یعنی زمان پاسخ مورد مقایسه قرار گرفت و نتایج نشان می دهد که روش پیشنهادی این معیار را نیز به خوبی بهبود بخشیده است که حاکی از در نظر گرفتن هر دو معیار به صورت توأم است. شکل ۳ نیز این مقایسه را به خوبی نشان می دهد.



شکل ۳. مقایسه زمان پاسخ روش‌های مختلف

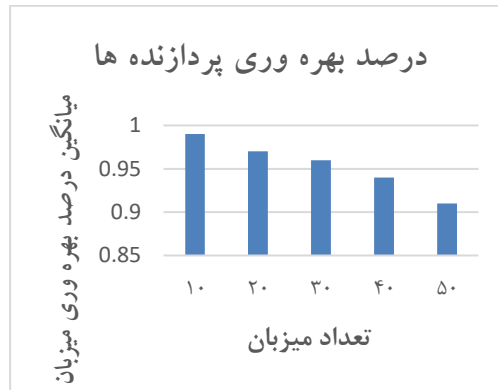
روش پیشنهادی، میانگین زمان پاسخ وظایف را نیز به خوبی در مقایسه با مطالعه پیشین بهبود می‌بخشد. در این روش علاوه بر صرفه‌جویی در هزینه‌ها، تعادل بار برای سیستم‌های محاسبات توزیع شده نیز کار می‌کند و برای ایجاد تعادل در وظایف ورودی دارای کارایی مناسب است. این الگوریتم زمان اجرا را به عنوان پارامتر اصلی کیفیت سرویس در نظر می‌گیرد و منجر به بهبود معیارهای کیفی مختلف می‌شود. نتایج به دست آمده در (شکل ۳) بیان‌گر بهره‌وری مناسب روش پیشنهادی در مقایسه با سایر روش‌ها است. نسبت زمان مشغول اجرا بودن یک منبع محاسباتی درون سیستم محاسبات ابری نسبت به کل زمان موجود را بهره‌وری یک منبع محاسباتی می‌نامند. در واقع، این معیار بیان‌گر نسبت استفاده مناسب از منابع است که مقدار آن عددی بین صفر و یک است. هر چه این عدد به یک نزدیک‌تر باشد بیان‌گر بهره‌وری بیش‌تر و هرچه کم‌تر از یک بوده و به صفر نزدیک باشد بیان‌گر اتلاف منابع محاسباتی است. با تغییر تعداد میزبان‌ها، حجم کارهای محول شده به هر میزبان تغییر می‌یابد، اما باید به این نکته توجه داشت که روش

پیشنهادی در شرایطی دارای کارآمدی مناسب می باشد که بتواند در تمامی شرایط موجود توازن بار مناسبی بین میزبان های ابری برقرار کند و حجم مناسبی از وظایف را بین آنها توزیع نماید. برای نشان دادن توازن مناسب از انحراف معیار متغیر بهره وری پردازنده های مختلف استفاده می شود. جدول ۳ کارایی روش پیشنهادی را در حالات مختلف نشان می دهد.

جدول ۳. بررسی میزان و انحراف معیار بهره وری پردازنده ها

تعداد میزبان ها	درصد بهره وری پردازنده ها	انحراف معیار
۱۰	۰/۹۹	۰
۲۰	۰/۹۷	۰/۰۲
۳۰	۰/۹۶	۰/۰۲
۴۰	۰/۹۴	۰/۰۳
۵۰	۰/۹۱	۰/۰۳

همان طور که نتایج ارائه شده در جدول ۳ نشان می دهد، روش پیشنهادی در تمامی حالات مختلف تعداد میزبان ها، بهره وری پردازنده را در نسبت بالای ۹۰ درصد برقرار کرده است که این بیانگر کیفیت مناسب روش ارائه شده در توزیع وظایف بین پردازنده ها است. همان طور که نتایج نشان می دهد با افزایش اعداد میزبان ها، با افت بهره وری پردازنده ها مواجه هستیم که البته با توجه به افزایش محاسباتی با افت ترافیک و کاهش وظایف تخصیصی به هر پردازنده روبرو خواهیم شد که این امر منجر به کاهش اندک بهره وری پردازنده ها خواهد شد. نمودار ارائه شده در شکل ۳ نیز این مقایسه را به خوبی نشان می دهد.



شکل ۴. نمودار بهره وری پردازنده ها

نمودار ارائه شده در شکل ۴ نیز به خوبی بهره وری مناسب روش پیشنهادی را نشان می دهد. علاوه بر این بررسی بیشتر جدول ۳ نشان می دهد که درصد انحراف معیار پردازنده ها در میزان بهره وری نیز بسیار اندک است و روش پیشنهادی به خوبی توانسته است بهره وری تقریباً یکسان را در بین پردازنده ها ایجاد کند و به خوبی شرایط پویایی مسئله را مدیریت کرده و نگاشت خوبی بین وظایف و میزبان های ابری برقرار می سازد.

نتیجه گیری و پیشنهادها

در این پژوهش رویکردی جدید بر مبنای الگوریتم بهینه سازی ازدحام ذرات چندهدفه برای بهینه کردن مسئله زمان بندی فعالیت ها در محیط توزیع شده محاسباتی بر حسب اولویت دهی به اجرای وظایف بر حسب هزینه های اجرایی بر روی ماشین و هزینه های تغییر ماشین ارائه شده است. پارامترهای هزینه ماشین و هزینه تغییر ماشین به طور هم زمان در روش پیشنهادی در نظر گرفته شده اند و سپس با ایجاد توازن میان فاکتورها، بر اساس میزان اهمیت کاربر برای هر پارامتر، به زمان بندی وظایف در محیط محاسبات ابری پرداخته شد. نتایج آزمایشی بر این موضوع دلالت داشته اند که الگوریتم پیشنهادی در مقایسه با دیگر مطالعه پیشین، بهتر عمل می کند. نتایج نشان می دهد که الگوریتم پیشنهادی هزینه های اضافی را به خوبی کاهش می دهد. این روش بر اساس تعادل بار برای سیستم های محاسبات توزیع ناهمگن کار می کند و برای ایجاد تعادل در وظایف دارای کارایی مناسبی می باشد.

منابع

- Bellavista, P., Montanari, R., & Das, S. K. (2013). Mobile social networking middleware: A survey. *Pervasive and Mobile Computing*, 9(4), 437-453.
- Chang, R. S., Chang, J. S., & Lin, P. S. (2009). An ant algorithm for balanced job scheduling in grids. *Future Generation Computer Systems*, 25(1), 20-27.
- Deshpande, U., You, Y., Chan, D., Bila, N., & Gopalan, K. (2014, June). Fast server deprovisioning through scatter-gather live migration of virtual machines. In *2014 IEEE 7th International Conference on Cloud Computing* (pp. 376-383). IEEE.
- Esmaili, Mohammad, Mirzaei, Abbas, 2015, Scheduling Tasks in Cloud Computing Using Enhanced Particle Swarm Optimization Algorithm, *International Conference on New Research Findings in Electrical Engineering and Computer Science*, *in persian*.
- Galloway, M., Loewen, G., & Vrbsky, S. (2015, June). Performance metrics of virtual machine live migration. In *2015 IEEE 8th International Conference on Cloud Computing* (pp. 637-644). IEEE.
- Garg, S. K., Yeo, C. S., Anandasivam, A., & Buyya, R. (2009). Energy-efficient scheduling of HPC applications in cloud computing environments. *arXiv preprint arXiv:0909.1146*.
- Gharooni-fard, G., Moein-darbari, F., Deldari, H., & Morvaridi, A. (2010). Scheduling of scientific workflows using a chaos-genetic algorithm. *Procedia Computer Science*, 1(1), 1445-1454.
- Huang, D., Yang, D., Zhang, H., & Wu, L. (2012, December). Energy-aware virtual machine placement in data centers. In *2012 IEEE Global Communications Conference (GLOBECOM)* (pp. 3243-3249). IEEE.
- Kamel-Tabakh Farizani, Seyyed Reza, Hashemi, Seyyed Tayebbeh, 2016, Algorithmic Load Balance Algorithm in Cloud Computing Inspired by Bee Behavior - *International Conference on Modern Research in Engineering Sciences*, *in persian*.
- Kumar, P., & Verma, A. (2012, August). Scheduling using improved genetic algorithm in cloud computing for independent tasks. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics* (pp. 137-142). ACM.
- Latiff, M. S. A., Abdul-Salaam, G., & Madni, S. H. H. (2016). Secure scientific applications scheduling technique for cloud computing environment using global league championship algorithm. *PLoS one*, 11(7), e0158102.
- Li, J., Peng, J., Lei, Z., & Zhang, W. (2011). An energy-efficient scheduling approach based on private clouds. *Journal of Information & Computational Science*, 8(4), 716-724.
- Lin, C. C., Jian, Z. D., & Hsu, C. H. (2014, December). A strategy of service quality optimization for live virtual machine migration. In *2014 IEEE 17th International Conference on Computational Science and Engineering* (pp. 1308-1313). IEEE.
- Lin, C. C., Jian, Z. D., & Hsu, C. H. (2014, December). A strategy of service quality optimization for live virtual machine migration. In *2014 IEEE 17th International Conference on Computational Science and Engineering* (pp. 1308-1313). IEEE.
- Liu, Y., Shao, H., Jing, W., & Qiu, Z. (2015). Multi-DAGs scheduling integrating with security and availability in cloud environment. *Chinese Journal of Electronics*, 24(4), 709-716.
- Oprescu, A. M., & Kielmann, T. (2010, November). Bag-of-tasks scheduling under budget constraints. In *2010 IEEE Second International Conference on Cloud Computing Technology and Science* (pp. 351-359). IEEE.
- Ramachandran, M., & Chang, V. (2016). Towards performance evaluation of cloud service providers for cloud data security. *International Journal of Information Management*, 36(4), 618-625.
- Sundar, S., & Liang, B. (2016, April). Communication augmented latest possible scheduling for cloud computing with delay constraint and task dependency. In *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHOPS)* (pp. 1009-1014). IEEE.
- Tawfeek, M. A., El-Sisi, A., Keshk, A. E., & Torkey, F. A. (2013, November). Cloud task scheduling based on ant colony optimization. In *2013 8th international conference on computer engineering & systems (ICCES)* (pp. 64-69). IEEE.
- Ungurean, I. (2010). Job scheduling algorithm based on dynamic management of resources provided by grid computing systems. *Elektronika ir Elektrotechnika*, 103(7), 57-60.

- Van den Bossche, R., Vanmechelen, K., & Broeckhove, J. (2013). Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds. *Future Generation Computer Systems*, 29(4), 973-985.
- Zhang, F., Chen, J., Chen, H., & Zang, B. (2011, October). CloudVisor: retrofitting protection of virtual machines in multi-tenant cloud with nested virtualization. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles* (pp. 203-216). ACM.
- Zhang, F., Chen, J., Chen, H., & Zang, B. (2011, October). CloudVisor: retrofitting protection of virtual machines in multi-tenant cloud with nested virtualization. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles* (pp. 203-216). ACM.
- Zhang, J., Ren, F., Shu, R., Huang, T., & Liu, Y. (2015). Guaranteeing delay of live virtual machine migration by determining and provisioning appropriate bandwidth. *IEEE Transactions on Computers*, 65(9), 2910-2917.

استناد به این مقاله:

شناسه دیجیتال (DOI): 10.22091/jemsc

قاسمی، راضیه؛ فاموری، فرزانه. (۱۳۹۷). «ارائه یک الگوریتم بهینه برای زمان‌بندی منابع و پارتیشن‌بندی کد در محاسبات ابری تلفن همراه». *مدیریت مهندسی و رایانش نرم*، ۷(۱)، ۲۴۴-۲۱۵.