دوره دوازدهم، شماره دوم، پاییز و زمستان ۱۳۹۸

مجله فناوری اطلاعات در طراحی مهندسی

Information Technology in Engineering Design

http://ited.sinaweb.net

یک روش جدید زمان‌بندی انرژی آگاه مبتنی بر AHP برای وظایف در محاسبات ابری

علی آقابابایی پور [*(۱)] شمس الله قنبری [(۱)]

(۱) گروه مهندسی کامپیوتر، واحد آشتیان، دانشگاه آزاد اسلامی، آشتیان، ایران.[*]

(تاریخ دریافت : ۱۳۹۹/۱/۱۹ تاریخ پذیرش : ۱۳۹۹/۵/۲)

چکیده

امروزه حجم داده‌های پردازشی توسط افراد و سازمان‌ها به‌طور چشم‌گیری افزایش‌یافته است که باعث افزایش روزافزون تقاضا برای پردازش‌های ابری می‌شود. الگوریتم‌های زمان‌بندی بسیاری ایجادشده که همگی سعی در کاهش زمان اجرای وظایف دارند و به کاهش مصرف انرژی توجهی نکرده‌اند. برخی دیگر از الگوریتم‌های زمان‌بندی سعی در کاهش زمان اجرا و انرژی مصرف‌شده به‌طور هم‌زمان دارند که الگوریتم‌های زمان‌بندی انرژی محور نامیده می‌شوند. هدف از الگوریتم ارائه‌شده در این مقاله زمان‌بندی وظایف با رویکرد کاهش مصرف انرژی و زمان اجرا، با استفاده از تصمیم‌گیری چند معیاره و یافتن بهترین گره برای اجرای وظایف می‌باشد. این الگوریتم بر اساس روش تصمیم‌گیری چند معیاره، بهترین گره برای اجرای وظیفه را پیشنهاد می‌دهد. پارامترهای مذکور شامل ارتباط بین وظایف، فاصله بین گره‌ها، وضعیت ماشین‌های مجازی و پیش‌بینی انرژی مصرفی می‌باشند. این روش با کاهش تعداد انتقالات و کاهش تغییر وضعیت‌های ماشین‌های مجازی توانسته است انرژی مصرفی را کاهش داده و هم‌چنین زمان اجرای را بهبود دهد. یافته‌های این مقاله نشانه داده که بهینه‌سازی راه‌اندازی ماشین‌های مجازی، بین ۳ تا ۵ درصد میزان مصرف انرژی را کاهش می‌دهد.

واژه‌های کلیدی: زمان‌بندی، انرژی محور، زمان اجرا، فرآیند تحلیل سلسله مراتبی، گراف جهت‌دار مدور

* عهده‌دار مکاتبات:

نشانی: گروه مهندسی کامپیوتر، واحد آشتیان، دانشگاه آزاد اسلامی، آشتیان، ایران.

پست الکترونیکی: *aghababaeipoor@gmail.com*

۱. مقدمه

محاسبات ابری تبدیل به یک ابزار بسیار مهم برای صنایع مرتبط با تکنولوژی و سازمان‌های فناوری اطلاعات شده است و بسیاری از مشکلاتی که در مسیر ارائه خدمات سریع و باکیفیت به مشتریان بوده را حل کرده است [۱]. سازمان‌های ارائه‌دهنده خدمات مایل هستند از روش‌هایی مانند pay-as-you-go یا elastic service در محیط‌های ابری استفاده کنند تا خدمات سریع‌تری برای مشتریان خود مهیا نمایند [۲]. با رشد سریع محاسبات ابری و افزایش مصرف انرژی توسط مراکز داده، محققان روش‌های متفاوتی را با محوریت کاهش مصرف انرژی ارائه کردند. در این روش‌ها تلاش شده است علاوه بر کاهش زمان اجرا، مصرف بهینه انرژی نیز صورت پذیرد. مانند روش سخت‌افزار مقیاس بندی پویای ولتاژ و فرکانس [۳] یا روش‌های نرم‌افزاری با تغییر در الگوریتم زمان‌بندی [٤] [٥]. برخی مطالعات نیز از روش تقسیم بار جهت زمان‌بندی وظایف استفاده می‌کنند که در آن بر اساس آن می‌توان بار را در تعداد زیادی پردازنده مستقل توزیع نمود [٦]. با توجه به نتایج ارائه‌شده در [۷] انتشار کربن ناشی از فعالیت‌های فناوری اطلاعات و ارتباطات سالیانه ٦٪ رشد دارد که بیشترین مصرف انرژی جهت انجام محاسبات در مراکز داده اتفاق می‌افتد و در حدود ٤٠٪ از کل انرژی مصرفی توسط مرکز داده است. سایر قسمت های مرکز داده مانند سیستم خنک‌کننده، سیستم تأمین برق ، ابزار ارتباطی و غیره، بخش دیگری هستند که انرژی مصرفی مرکز داده را تشکیل می‌دهند. یکی از روش های مؤثر کاهش انرژی مصرفی در محیط‌های ابری استفاده از الگوریتم های زمان‌بندی مبتنی بر کاهش انرژی است به‌صورت که فاکتور کاهش انرژی موجب تأثیر نامطلوب بر زمان اجرا نشود. درروش ارائه‌شده، تعداد انتقالاتی که هر وظیفه باید داشته باشد تا به گره منتخب جهت اجرا برسد به‌عنوان یکی از پارامترهای موردنیاز جهت بهبود زمان اجرا و مصرف انرژی در نظر گرفته می شود . پارامتر دیگر ارتباطات بین وظایف است که از گراف جهت‌دار غیرمدور داده شده بدست می‌آید. گراف جهت‌دار غیرمدور شامل وظایف ورودی و ارتباطات بین آن‌هاست. همچنین وضعیت ماشین‌های مجازی به‌عنوان پارامتر دیگر موردبررسی قرار می گیرد تا از خاموش و روشن شدن‌های غیرضروری جلوگیری شود. در آخر انرژی موردنیاز هر وظیفه بر اساس روش ارائه‌شده محاسبه می شود و با استفاده از این چهار پارامتر و روش تصمیم‌گیری تحلیل سلسله مراتبی، بهترین گره برای اجرای وظیفه انتخاب می‌گردد.

الگوریتم ارائه‌شده به‌عنوان هدف اصلی تلاش دارد تا تعداد انتقالات وظایف را کاهش دهد و در مرحله بعد از خاموش و روشن شدن‌های ماشین‌های مجازی تا حد امکان جلوگیری نماید. کاهش این دو خصوصیت تأثیر مستقیم بر زمان اجرا و مصرف انرژی وظایف دارد. مقاله تلاش دارد به این سؤال پاسخ دهد که می‌توان از راه‌اندازی‌ها و خاموش شدن‌های مکرر ماشین‌های مجازی جلوگیری کرد و همچنین چگونه می‌توان تعداد انتقالات بین گره‌ها را کاهش داد. دستاوردهای این تحقیق به شرح ذیل می‌باشد:

۱. در این مقاله یک الگوریتم زمان‌بندی انرژی محور ارائه‌شده است. این الگوریتم که چند پارامتری می‌باشد، بهترین انتخاب را در هر دور جهت اجرای وظایف، با رویکرد کاهش انرژی و افزایش سرعت اجرا انجام می‌دهد.

۲. شبیه‌سازی و مقایسه‌های عددی ارائه‌شده است تا بهبود سرعت اجرا و کاهش مصرف انرژی تائید نماید. این نتایج، میزان تأثیر مدیریت ماشین‌های مجازی بر کاهش مصرف انرژی را نشان می‌دهد.

۳. در این مقاله، فاکتورهایی که بر عملکرد الگوریتم تأثیرگذار هستند موردبررسی قرارگرفته‌اند. ساختار این مقاله به‌صورت که در ادامه ذکر می‌شود، است. در بخش ۲ اشاره مختصری به پیشینه تحقیقات انجام‌شده در خصوص زمان‌بندی وظایف با رویکرد کاهش انرژی، شده است. در بخش ۳، به تشریح جزییات مدل‌سازی سیستم پرداخته‌شده است. در ادامه و در بخش ٤، مدل تصمیم‌گیری الگوریتم زمان‌بندی را توضیح داده‌ایم و درنهایت در بخش ۵ مقاله، نحوه شبیه‌سازی و نتایج بدست آمده ارائه‌شده است.

۲. پیشینه تحقیق

افزایش مصرف انرژی در سیستم‌های کامپیوتری با مقیاس بزرگ مانند ابرها، پیامدهای اقتصادی و زیست‌محیطی زیادی را به دنبال دارد. ازاین‌رو، کاهش مصرف انرژی یک موضوع پیچیده و چالش‌برانگیز در طراحی سیستم‌های مدرن است، به دلیل اینکه سرعت رشد داده‌ها و برنامه‌های کاربردی محاسباتی رو به افزایش است، بیش از قبل، لزوم وجود سرویس‌دهنده‌ها و دیسک‌ها مشخص می‌شود، تا این داده‌ها و برنامه‌های کاربردی را با سرعت کافی و در بازه زمانی موردنیاز، پردازش کنند. بنابراین باید تمرکز از بهینه‌سازی مدیریت منابع مرکز داده صرفاً برای کار آیی به سمت بهینه‌سازی آن‌ها برای بازدهی انرژی تغییر پیدا کند و در ضمن، کار آیی بالا در سطح ارائه سرویس حفظ شود. ازاین‌رو، فراهم‌کنندگان سرویس ابری بایستی سیاست‌هایی را اتخاذ کنند تا تضمین کنند که مزایای استفاده از آن‌ها، به دلیل هزینه‌های بالای انرژی به طرز چشمگیری کاهش نمی‌یابد [۸]. بنابراین یکی از چالش‌های مهم در محاسبات ابری، مصرف انرژی است. برای کاهش مصرف انرژی الگوریتم‌ها و ساختارهای مختلفی برای محاسبات ابر پیشنهاد شده است. هریک سعی به کاهش مصرف انرژی و بهره‌برداری بهتر از منابع رادارند. در این تحقیق به مطالعه و ارزیابی روش‌های کاهش مصرف انرژی در محاسبات ابری پرداخته‌شده است.

تحقیق [۹] تمرکز خود را بر زمان‌بندی ماشین مجازی پویا معطوف می‌کنند برای دستیابی به بهره‌وری انرژی و تأمین محدودیت‌های مهلت با PM های مختلف در ابر. تعداد کافی از PM ها برای پردازش تعداد مشخصی از ماشین مجازی با فرکانس بهینه وجود دارد و هر PM روی حداقل فرکانس بهینه عمل می‌کند. بهترین عملکرد قدرت موردنیاز PM ها از ایده استفاده از دستگاه‌های مختلف فیزیکی بدست می‌آید، جایی که ماشین مجازی نسبت قدرت را قبل از PMها برای افزایش عملکرد اختصاص می‌دهد. در محاسبات ابری، زمان‌بندی به رویدادهای زمانی کوچک تقسیم می‌شود تا میزان مصرف انرژی کاهش یابد. رویداد زمان مهلت برای مدیریت منابع ماشین مجازی از PM ها لازم است. اطلاعات مربوط به زمان پاسخ میانگینِ مسئله برنامه‌نویسی عددی در مرکز داده برای تخصیص وظایف وجود دارند، که در سرویس‌دهنده‌های پویا محدودشده‌اند [۱۰]. سرویس‌دهنده‌های پویا در مرکز داده انرژی بیشتری مصرف می‌کنند و در زمان‌بندی کار آیی ندارند. درحالی‌که طرح‌های حداکثر صرفه‌جویی در انرژی برای سرویس‌دهنده‌های مرکز داده در طرح زمان‌بندی وظیفه از کارآمدترین اولین سرویس‌دهنده استفاده می‌شود. با استفاده از این طرح، متوسط زمان پاسخ کار کاهش می‌یابد. همچنین این انرژی را در هزینه زمان‌های پاسخ‌دهی کارها محافظت می‌کند، گرچه در محدوده ماکزیمم. تحقیق [۱۱] یک الگوریتم زمان‌بندی وظایف انرژی آگاه هوشمند پیشنهاد دادند با قصد افزایش سطح بهینه استفاده با پیشنهاد مازاد انرژی ماشین مجازی به ارائه‌دهندگان ابر. الگوریتم پیشنهادی سرویس‌دهنده‌هایی را ایجاد می‌کند تا ماشین‌های مجازی

خود را سریع‌تر برای دستیابی به سطوح بهینه‌سازی خود بدون نیاز به انتقال ماشین مجازی پیدا کنند. به‌این‌ترتیب، سرویس‌دهنده‌های غیرفعال برای صرفه‌جویی در انرژی خاموش می‌شوند. آزمایش‌ها نشان می‌دهد که سیاست پیشنهادشده نه‌تنها مصرف انرژی را کاهش می‌دهد، بلکه از سوی دیگر باعث می‌شود که زمان پردازش کل وظایف بلادرنگ در ابر کاهش می‌یابد.

بررسی جامع الگوریتم بهینه‌سازی ازدحام ذرات و کاربرد در محاسبات ابر با استفاده از معماری سیستم CloudSim در [۱۲] ارائه‌شده است. مسئله زمان‌بندی منابع به‌صورت جامع در نظر گرفته‌شده است و به عملکرد، مقیاس‌پذیری و قابلیت اطمینان دست می‌یابد. به‌عنوان یک نقطه نهایی، درخواست کاربران مرتبط با منطقه محاسبات ابری یافت می‌شود. سه پارامتر برای تعیین مقادیر به‌صورت پویا با استفاده از PSO به‌عنوان استراتژی زمان‌بندی منابع در محاسبات ابر موردبررسی قرار می‌گیرند. نتایج آزمایش نشان می‌دهد که رویکرد زمان‌بندی منابع پیشنهادشده، استفاده از انرژی را کاهش می‌دهد و عملکرد مرکز داده را افزایش می‌دهد؛ این استراتژی تأثیر بسیار مثبتی بر محاسبات ابری دارد.

تحقیق [۱۳] مسئله دشوار جایگذاری ماشین مجازی در مراکز ابری را بررسی کردند. کار عمده آن‌ها استفاده از الگوریتم‌های جایگذاری و تلفیق است که هزینه‌های مهاجرت و مصرف انرژی را کاهش می‌دهد. آن‌ها یک برنامه عددی خطی را که برای توزیع دقیق الگوریتم مهاجرت ماشین مجازی طراحی‌شده است، به‌منظور به حداقل رساندن مصرف انرژی از طریق اتحاد، پیشنهاد کردند. تحقیق [۱٤] الگوریتم‌های تخصیص منابع انرژی آگاه و الگوریتم‌های زمان‌بندی را بررسی کردند که منابع ابر را به کاربران ابر تخصیص می‌دهد که باعث افزایش بهره‌وری انرژی یک مرکز داده بدون ایجاد اختلال در SLA می‌شود. برای دستیابی به کار آیی انرژی، مکانیزم‌های خودگردان و انرژی آگاه برای خود مدیریتی استفاده می‌شود. الگوریتم‌های پیشنهادی برای نگاشت کارآمد انرژی از ماشین‌های مجازی برای منابع ابر مناسب است. نتایج شبیه‌سازی نشان می‌دهد که روش پیشنهادی مصرف انرژی در مراکز داده ابری را کاهش می‌دهد.

تحقیق [۱۵] یک سیاست زمان‌بندی منابع محاسبات ابری در الگوریتم ژنتیک با پردازش عددی را معرفی می‌کنند، به‌منظور بهبود استفاده از منابع و صرفه‌جویی در هزینه‌های انرژی در محاسبات ابری. تحقیق [۱۵] از یک رویکرد پیش مهاجرت بر اساس سه درجه بار استفاده می‌کنند مانند استفاده از پردازنده، توان عملیاتی شبکه، نرخ I / O دیسک، که در الگوریتم مربوطه اندازه‌گیری می‌شوند. به‌منظور بدست آوردن یک‌راه حل تقریباً مطلوب، الگوریتم ژنتیک با مسئله کوله‌پشتی از طریق برازش‌های مختلف ترکیب شد و نتایج نشان‌دهنده بهره‌وری الگوریتم پیشنهادی هستند. الگوریتم موفق به بالا بردن بهره‌برداری از منابع و صرفه‌جویی در هزینه‌های انرژی در زمان اجرا شد. در این مقاله، [۱٦] بهره‌وری انرژی در محاسبات ابری را توصیف کردند. مدل مصرف انرژی ارائه‌شده است، و انواع منابع محاسباتی به چهار دسته مختلف تقسیم‌بندی شدند که شامل پردازنده، حافظه، ذخیره‌سازی، و شبکه است. علاوه بر این، طراحی مقررات و استراتژی‌های مختلف برای اجزای متعدد نیز صورت گرفته است. علاوه بر این، این مقاله یک الگوریتم زمان‌بندی منابع پویا بر اساس بهینه‌سازی مصرف انرژی منابع ابر را با روش ارزیابی پیشنهاد می‌کند. نتایج شبیه‌سازی نشان‌دهنده عملکرد بهتر الگوریتم ازنظر مصرف انرژی است. مدل عملکرد سیستماتیک مراکز داده پشتی از طریق برازش‌های مختلف ترکیب شد و نتایج نشان‌دهنده بهره‌وری الگوریتم پیشنهادی هستند. الگوریتم موفق به

بالا بردن بهره‌برداری از منابع و صرفه‌جویی در هزینه‌های انرژی در زمان اجرا شد. در این مقاله [۱۶]، بهره‌وری انرژی در محاسبات ابری را توصیف کردند. مدل مصرف انرژی ارائه‌شده است، و انواع منابع محاسباتی به چهار دسته مختلف تقسیم‌بندی شدند که شامل پردازنده، حافظه، ذخیره‌سازی، و شبکه است. علاوه بر این، طراحی مقررات و استراتژی‌های مختلف برای اجزای متعدد نیز صورت گرفته است. علاوه بر این، این مقاله یک الگوریتم زمان‌بندی منابع پویا بر اساس بهینه‌سازی مصرف انرژی منابع ابر را با روش ارزیابی پیشنهاد می‌کند. نتایج شبیه‌سازی نشان‌دهنده عملکرد بهتر الگوریتم ازنظر مصرف انرژی است. مدل عملکرد سیستماتیک مراکز داده توسط [۱۷] ارائه‌شده است. این الگوریتم کیفیت خدمات کیفیت سرویس را به کاربران و استفاده کاربردی انرژی در مراکز داده را در نظر می‌گیرد. به غیر از انرژی مصرف بالقوه در محدوده عملکردی با توزیع ماشین‌های مجازی به PMS بر اساس اولویت به نتایج ریاضی ارتباط استراتژی تخصیص مختلف بستگی دارد.

عنوان تحقیق	روش	معیارها	مزایا	معایب
تحقیق ۱۱	مدیریت ماشین های مجازی غیرفعال	وضعیت ماشین های مجازی	کاهش انرژی مصرفی کاهش زمان کل	تعویق احتمالی در وظایف با اولویت بالا
تحقیق ۱۲	مبتنی بر الگوریتم PSO	انرژی مصرفی منابع در اختیار زمان اجرا	کاهش مصرف انرژی افزایش عملکرد جامع مرکز داده	به موضوع بهینه سازی سرعت اجرا پرداخته نشده است.
تحقیق ۱۳	الگوریتم جایگذاری و تلفیق	انرژی مصرفی وضعیت ماشین مجازی	کاهش هزینه مهاجرت ماشین مجازی کاهش مصرف انرژی	به موضوع بهینه سازی سرعت اجرا پرداخته نشده است.
تحقیق ۱۴	مکانیزم خودگردان و انرژی آگاه	منابع مرکز داده انرژی مصرفی منابع	کاهش مصرف انرژی	به موضوع بهینه سازی سرعت اجرا پرداخته نشده است.
تحقیق ۱۵	رویکرد پیش مهاجرت بر اساس سه درجه بار با کمک الگوریتم ژنتیک	درجه بار مهاجرت ماشین های مجازی انرژی مصرفی زمان اجرا	بالا بردن بهره برداری از منابع صرفه جویی در هزینه‌های انرژی	بهبودهای بدست آمده برای بخشی از وظایف قابل تعمیم است
تحقیق ۱۶	بررسی مصرف انرژی زیرسیستم ها	مصرف انرژی پردازنده، حافظه، شبکه	کاهش مصرف انرژی	به موضوع بهینه سازی سرعت اجرا پرداخته نشده است.
تحقیق ۱۷	مدیریت کیفیت ارائه خدمات در ابر	کیفیت ارائه خدمات انرژی مصرفی	بهبود ارائه خدمات کاهش انرژی	کاهش انرژی به صورت موردی رخ می دهد و جامع نیست، به بهبود زمان اجرا پرداخته نشده است.

جدول ۱ متغیرها و نمادها

۳. مدل‌سازی سیستم

پارامترهای استفاده شده در الگوریتم پیشنهادی در جدول ۱ نشان داده شده است. الگوریتم ارائه‌شده با استفاده از چهار پارامتر شامل ارتباط بین وظایف، تعداد انتقالات موردنیاز جهت انتقال وظیفه به گره مربوطه، وضعیت ماشین‌های مجازی هر گره و میزان انرژی مصرف‌شده برای اجرای وظیفه، زمان‌بندی وظایف وارده از گراف جهت‌دار بدون دور را انجام می‌دهد. الگوریتم زمان‌بندی در قدم اول تعداد انتقالات بین گره‌ها را کاهش می‌دهد به‌گونه‌ای که وظایف مرتبط در گره‌های هم‌جوار اجرا شوند تا در سیکل‌های بعدی تعداد انتقالات کمتری برای رسیدن وظایف مرتبط به یکدیگر

صورت پذیرد. در قسمت دوم الگوریتم زمان‌بندی، وضعیت ماشین‌های مجازی مانیتور می‌شود تا وظایف توسط ماشین‌های مجازی آنلاین اجرا شوند و از خاموش و روشن کردن ماشین‌های مجازی تا حد امکان اجتناب شود. به‌عنوان کنترل نهایی، الگوریتم با استفاده از پروفایل انرژی ارائه‌شده در [۱۸] میزان مصرف انرژی برای اجرای وظیفه ورودی را محاسبه می‌کند تا درنهایت انتخابی متوازن بین زمان اجرا و مصرف انرژی برای اجرای وظیفه داشته باشد.

به‌عبارت‌دیگر این الگوریتم به دنبال نزدیک‌ترین گره‌ای است که بیشترین تعداد وظایف مرتبط با وظیفه ورودی را شامل می‌شود و ماشین‌های مجازی این گره نیازی به راه‌اندازی نداشته باشند و کمترین مصرف انرژی برای اجرای وظیفه ورودی در آن حالت رخ دهد. برای مثال ممکن است الگوریتم گره ای را انتخاب کند که فاصله بیشتری تا گره فعلی دارد اما نیازی به راه اندازی ماشین مجازی جدید در آن گره نباشد. این انتخاب به کمک روش تحلیل سلسله مراتبی و بر اساس اولویت‌های تعریف‌شده صورت می‌پذیرد. به‌صورت که الگوریتم بر

اساس اولویت‌های ذکرشده مناسب‌ترین مکان برای اجرای وظیفه ورودی با کمترین مصرف انرژی و زمان اجرا را انتخاب می‌کند. ازآنجایی‌که برای محاسبه پارامترهای ذکرشده نیاز به منابع پردازشی مجزا و همین‌طور فضای ذخیره-سازی جهت کَش کردن برخی اطلاعات است لذا یک ماشین مجازی به‌عنوان هماهنگ‌کننده در نظر گرفته می‌شود تا الگوریتم زمان‌بندی در آن اجرا گردد. شمای کلی روش ارائه‌شده در شکل (۱) نشان داده شده است.

داده‌های ورودی ابتدا تبدیل به گراف جهت‌دار بدون دور می‌شوند تا ارتباط بین وظایف مشخص شود. گراف جهت‌دار بدون دور که به‌صورت G=(T,D) نشان داده می‌شود شامل رأس و یال می‌باشد، رأس‌ها نشان‌دهنده وظایف هستند و یال-ها ارتباط بین وظایف را نشان می‌دهند. برای مثال i->j نشان می‌دهد که Taskj برای اجرا نیاز به Taski دارد. بنابراین i و j هرکدام یک رأس و i->j معادل یک یال در گراف جهت‌دار بدون دور است. این گراف به‌عنوان مجموعه وظایف ورودی به محیط ابری فرستاده می‌شود. محیط ابری شامل مجموعه‌ای از گره‌های پردازشی در نقاط مختلف با فواصل مختلف است که هر گره خود از تعدادی ماشین‌مجازی روشن یا خاموش تشکیل‌شده است. لیست گره‌های موجود در ابر به‌صورت (N=(n1, n2, …, nm نشان داده می‌شود که m نشان‌دهنده تعداد گره‌های ابر است. لیست ماشین‌های مجازی روشن در گره m به‌صورت VMonm=(VMonm1, VMonm2, …,) VMonmk و لیست ماشین‌های مجازی خاموش در گره m به‌صورت VMoffm = (VMoffm1, VMoffm2, …, VMoffml) نشان داده می‌شود که k نشان‌دهنده تعداد ماشین‌های مجازی روشن در گره m و l نشان‌دهنده تعداد ماشین‌های مجازی خاموش در گره m است. همچنین در محیط ابری ارائه‌شده VMcoordinator نشان‌دهنده ماشین مجازی هماهنگ‌کننده است که وظیفه اجرای الگوریتم زمان‌بندی را به عهده دارد. الگوریتم زمان‌بندی S برای هر وظیفه ورودی ti نحوه تخصیص را بر اساس پارامترهای معرفی‌شده پیدا می‌کند. الگوریتم زمان‌بندی S جهت کاهش مصرف انرژی در هنگام تخصیص وظایف از پروفایل انرژی ارائه‌شده در [۱۸] استفاده می‌کند.

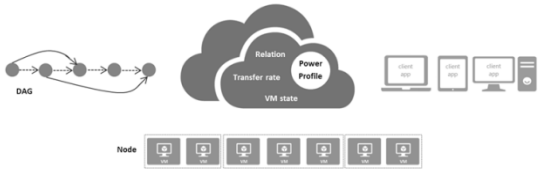
همان‌طور که گفته شد الگوریتم ارائه‌شده جهت زمان‌بندی وظایف از چهار پارامتر استفاده می‌کند که در این بخش توضیح داده‌شده است.

ارتباط بین وظایف: این مقدار از تشکیل گراف جهت‌دار بدون دور به دست می‌آید. در این گراف مشخص‌شده هر وظیفه ورودی ti با کدام‌یک از وظایف دیگر که تاکنون اجراشده‌اند در ارتباط است. الگوریتم زمان‌بندی به هر گره بر اساس تعداد وظایف مرتبطی که با ti دارد اولویت می‌دهد تا ti در گره‌ای که بیشترین وظایف مرتبط با آن وجود دارند اجرا شود. این امر از جابه‌جایی‌های زیاد در مراحل بعدی پردازش وظایف جلوگیری می‌کند و موجب کاهش زمان اجرا و مصرف انرژی می‌گردد. فرمول ۱ روش محاسبه Relti را نشان می‌دهد:

$$\mathrm{Rel}_{t_i} = \{\mathrm{Rel}_{t_1}, \mathrm{Rel}_{t_2}, \dots, \mathrm{Rel}_{t_m}\} \qquad (۱)$$

که m تعداد کل گره‌هاست و ارائه ارتباط بین وظایف در هر گره توسط فرمول ۲ محاسبه می‌شود:

$$\mathrm{Rel}_{t_k} = \sum_{n=0}^{i-1} \mathrm{relation}(t_n, t_i) \qquad (۲)$$



نرخ انتقالات: پس‌ازآنکه الگوریتم زمان‌بندی گره‌های مرتبط با ti را مشخص کرد، تعداد انتقالاتی که لازم است تا ti به هرکدام از این گره‌ها برسد را نیز محاسبه

می‌کند و بر اساس فواصل به‌دست‌آمده به هر گره‌یک اولویت می‌دهد. درواقع الگوریتم زمان‌بندی تلاش می‌کند با محاسبه انتقالات موردنیاز برای اجرای وظیفه، گره‌ای که کم‌ترین فاصله را دارد برای اجرای ti انتخاب کند. فرمول ۳ نشان می‌دهد که Transti بر اساس Relti محاسبه می‌شود:

$$\mathbf{Trans_{t_i}} = \left\{\mathbf{Trans_{t_{i_1}}}, \mathbf{Trans_{t_{i_2}}}, \dots, \mathbf{Trans_{t_{i_m}}}\right\} \quad (۳)$$

که m تعداد گره‌های مرتبط است و میزان انتقال برای رسیدن به هر یک از گره‌های مرتبط توسط فرمول ٤ محاسبه می‌شود:

$$\mathrm{Trans_{t_{i_k}}} = \mathrm{distance}(n_i, n_k) \quad (٤)$$

وضعیت ماشین مجازی: اولویت اجرای وظایف با ماشین‌های مجازی آنلاین است و الگوریتم زمان‌بندی تلاش می‌کند تا حد امکان از راه‌اندازی ماشین‌های مجازی آفلاین جلوگیری نماید. به این منظور لیست گره‌های مرتبط با ti موردبررسی قرار می‌گیرد تا وضعیت آن‌ها ازلحاظ تعداد ماشین‌های مجازی آنلاین و آفلاین بررسی شود و اجرای ti به گره‌ای سپرده شود که بیشترین احتمال آزاد بودن ماشین مجازی آنلاین رادار است. حالت ماشین‌های مجازی متغیری لحظه‌ای است که برای مثال ممکن است در زمان بررسی گره‌ها، گره‌ای شامل دو ماشین مجازی آزاد باشد اما تا زمانی که زمان‌بند ti را برای آن گره ارسال نماید این دو ماشین مجازی توسط بخش‌های دیگر ابر به کار گرفته شوند. برای حل این مشکل الگوریتم زمان‌بندی سه سیاست متفاوت را اتخاذ می‌کند. اول اینکه در پارامتر VMState گره‌هایی که بیشترین ماشین مجازی آنلاین آزاد رادارند اولویت بالاتری می‌گیرند تا شانس اجرای ti افزایش یابد. علاوه بر آن الگوریتم زمان‌بندی دو گره برای اجرای ti انتخاب می‌کند، گره اصلی بهترین انتخاب ممکن و گره دوم نزدیک‌ترین انتخاب به گره‌اصلی است تا در صورت اجرا نشدن احتمالی ti در گره اصلی، گره دوم اجرای آن را بر عهده گیرد. سیاست سومی که الگوریتم از آن برای افزایش ضریب اطمینان VMState استفاده می‌کند ساخت یک حافظه پنهان در هماهنگ‌کننده به نام Trust است. زمان‌بند پس از هر بار تخصیص وظیفه, وضعیت اجرای آن را بررسی می‌نماید، وظیفه در هر گره‌ای که اجرا شود زمان‌بند امتیاز آن گره را در متغیر Trust افزایش می‌دهد تا در دورهای بعدی پردازش، زمان‌بند جهت تشکیل VMState اولویت گره‌هایی که امتیاز بالاتری در Trust دارند را افزایش دهد. نحوه تشکیل VMState در فرمول ۵ نشان داده شده است.

$$\mathrm{VMstate} = \left\{\mathrm{evaluate}(\mathrm{VM_{on_m}}, \mathrm{VM_{off_m}}, \mathrm{Trust_m})\right\} \quad (۵)$$

مصرف انرژی: الگوریتم زمان‌بندی از پروفایل انرژی ارائه‌شده در [۱۷] جهت تخمین انرژی مصرفی برای اجرای $\mathbf{t_i}$ در هرکدام از گره‌های موجود در $\mathbf{Rel_{t_i}}$ استفاده می‌کند. این پروفایل، برای هر جزء از ابر انرژی مصرفی را به‌طور جداگانه محاسبه می‌کند که شامل انرژی مصرفی برای انتقال بین گره‌ها، انرژی لازم برای راه‌اندازی ماشین مجازی و پردازش وظیفه توسط آن، انرژی مصرف‌شده توسط وظیفه و همین‌طور گره می‌باشد. زمان‌بند ابتدا با اعمال پارامترهای قبلی مصرف انرژی را به حداقل می‌رساند سپس برای اطمینان از کاهش بیشتر انرژی، برای هر گره انرژی مصرفی را به‌طور جداگانه محاسبه می‌کند. الگوریتم زمان‌بندی بیشترین اولویت را به گره‌ای اختصاص می‌دهد که مجموع انرژی مصرف‌شده برای اجرای $\mathbf{t_i}$ در آن گره از سایر گره‌ها کم‌تر است. فرمول ٦ محاسبه انرژی برای $\mathbf{t_i}$ را نشان می‌دهد.

$$\mathrm{e_{total_t}} = \mathrm{Min}\left\{\mathrm{e_{total_{t_1}}}, \mathrm{e_{total_{t_2}}}, \dots, \mathrm{e_{total_{t_m}}}\right\} \quad (٦)$$

که m تعداد گره‌های مرتبط است و انرژی مصرف‌شده توسط هر وظیفه با فرمول ۷ محاسبه می‌شود:

$$e_{total_t} = e_{t_k} + e_{transfer\{t_t \rightarrow n_k\}} + e_{v_k} + e_{n_k} + e_{etc(cooling,\dots)} \quad (۷)$$

٤. مدل تصمیم‌گیری

الگوریتم زمان‌بندی توسط یک ماشین مجازی هماهنگ‌کننده اجرا می‌شود که برای تخصیص وظایف ورودی بین گره‌های موجود در ابر تصمیم‌گیری می‌کند. الگوریتم ۱ نحوه‌ی زمان‌بندی ارائه‌شده را توسط پارامترهای محاسبه‌شده در قسمت قبل را نشان می‌دهد. با استفاده از روابط بین وظایف که توسط گراف جهت‌دار بدون دور ایجادشده است، الگوریتم زمان‌بندی ارائه **relatedNodes** را تشکیل می‌دهد. این ارائه شامل تمام گره‌هایی می‌شود که حداقل یک وظیفه مرتبط با t_i در آن وجود داشته باشد و اولویت بالاتر به گره‌ای داده می‌شود که بیشترین وظیفه مرتبط با t_i رادار است .

الگوریتم زمان‌بندی ارائه **transfers** را برای ذخیره فاصله گره کنونی که t_i در آن قرارداد با هر یک از گره‌های ارائه **relatedNodes** تشکیل می‌دهد. این ارائه مشخص می‌کند کدام گره‌ی مرتبط با t_i کم‌ترین فاصله از آن را دارد تا زمان اجرا و مصرف انرژی ناشی از انتقال بین گره‌ها کاهش یابد. ارائه **energyUsage** نیز انرژی مصرفی برای اجرای t_i در هر یک از گره‌های **relatedNodes** را ذخیره می‌کند. پروفایل انرژی برای t_i و گره‌های مرتبط آن فراخوانی می‌شود تا با محاسبه انرژی مصرفی هر جزء از ابر مقدار کل انرژی ناشی از اجرای t_i به دست آید و گره‌ای که کم‌ترین مصرف انرژی در آن رخ می‌دهد مشخص می‌شود.

```
Procedure Scheduling
Input: incoming tasks to the cloud
Output: execution of task status
Begin
relatedNodes = CheckTaskRelationInCash( t_i)
transfer = Distance(n_ti, relatedNodes)
monitoredVM = monitorVM(relatedNodes)
energyUsag = PowerProfile(t_i, relatedNodes,
      transfer, monitoredVM)
weightOfNodes = AHP(relatedNodes, transfer,
      monitoredVM, energyUsage)
selectedNode = Max(weightOfNodes)
backupNode = nearBy(selectedNode)
trust = Execute(t_i, selectedNode, backupNode)
End
```

جهت بررسی وضعیت ماشین‌های مجازی و انتخاب گره‌ای با مناسب‌ترین وضعیت ماشین مجازی، الگوریتم زمان‌بندی ارائه‌ی monitoredVM را با اجرا کردن الگوریتم ۲ تشکیل می‌دهد. تابع monitorVM تمامی گره‌های موجود در ارائه‌ی relatedNodes را بررسی می‌کند، اگر ماشین مجازی آنلاین در آن گره موجود باشد، خانه متناظر آن گره در ارائه monitoredVM برابر تعداد ماشین‌های مجازی آنلاین آن گره مقداردهی می‌شود، در غیر این صورت این مقدار برابر تعداد ماشین‌های مجازی خاموش این گره خواهد بود و درصورتی‌که تمام ماشین‌های مجازی این گره در حال کار باشند مقدار null برای آن ثبت خواهد شد.

الگوریتم ۲ الگوریتم بررسی وضعیت ماشین مجازی

```
Procedure monitorVM
Input: relatedNodes array contain incoming tasks
Output: monitoredVM array contain VM status in
  each node
Begin
for each node in relatedNodes do
  if there is any online VM in
    relatedNodes {node}
    monitoredVM {node} = Count(VM_ON {node})
  end if
  if there is any off-line VM in
    relatedNodes {node}
    monitoredVM {node} = Count(VM_OFF {node})
  end if
  if there is't any off-line or online VM in
    relatedNodes {node}
    monitoredVM {node} = null
   end if
end for
End
```

پس‌ازآنکه پارامترهای موردنیاز الگوریتم بدست آمدند تصمیم‌گیرنده **AHP** فرخوانی می‌شود، تا وزن هر گره را بر اساس چهار پارامتر بدست آمده محاسبه نماید. وزن هر گره در ارائه‌ی **weightOfNodes** ذخیره می‌شود، زمان‌بند بیشترین وزن را به‌عنوان گره اصلی جهت اجرای t_i انتخاب می‌کند و در **selectedNode** ذخیره می‌نماید.

در ادامه با استفاده از تابع **nearby** نزدیک‌ترین گره به گره اصلی به‌عنوان گره پشتیبان انتخاب می‌شود تا در صورت پر شدن ماشین‌های مجازی گره اصلی بتواند در کوتاه‌ترین زمان و با صرف کمترین انرژی t_i را اجرا نماید. پس از اجرای t_i امتیاز گره اجراکننده در **Trust** افزایش می‌یابد تا در فراخوانی‌های بعدی مورداستفاده قرار گیرد.

۵. انتخاب بهترین گره توسط تحلیل سلسله مراتبی

تحلیل سلسله مراتبی برای تصمیم‌گیری از مقایسه زوج به زوج استفاده می‌کند و هر یک از گره‌ها را به‌صورت دوتایی برای هر پارامتر به‌صورت جداگانه محاسبه می‌نماید درنتیجه چهار ماتریس مقایسه‌ای هفت‌در هفت خواهیم داشت که به همراه ماتریس مقایسه‌ای پارامترها، نرمال شده و وزن آن‌ها محاسبه می‌شود. وزن‌های به‌دست‌آمده به وزن کل برای هر گره تبدیل می‌شوند و گره‌ای که بیشترین وزن رادار است به‌عنوان گره اصلی جهت اجرای وظیفه ورودی انتخاب می‌شود.

پس از تشکیل ماتریس‌های زوجی از پارامترها، اوزان مربوط به هر پارامتر به کمک فرمول ۸ محاسبه گردید.

$$a_{n1}w_1 + a_{n2}w_2 + \cdots + a_{nn}w_n = \lambda . w_n \qquad (۸)$$

سپس وظایف موجود در صف با توجه به اهمیت پارامترها که در فرمول ۸ محاسبه شد، وزن دهی می‌شوند و وظیفه‌ای که بیشترین اولویت جهت اجرا را داشته باشد در ابتدای صف اجرا قرار می‌گیرد.

٦. راه‌اندازی و تنظیمات

جهت شبیه‌سازی الگوریتم پیشنهادی، از مجموعه‌ای شامل دو تا ده گره استفاده شده است که هر گره متشکل از شش ماشین مجازی است.یک ماشین مجازی نیز به‌عنوان هماهنگ‌کننده انتخاب گردید، هماهنگ‌کننده ابتدا و قبل از شروع زمان‌بندی سرویس‌های موردنظر را راه‌اندازی می‌نماید. هماهنگ‌کننده وظایفی ورودی را به شکل گراف جهت‌دار بدون دور دریافت می‌کند، جهت مقایسه الگوریتم ارائه‌شده، شش گراف جهت‌دار بدون دور متفاوت مشتمل بر وظایف و ارتباط بین آن‌ها برای هماهنگ‌کننده ارسال شد که الگوریتم ارائه‌شده برای حالت‌های مختلف ارتباط بین وظایف تست شود. هماهنگ‌کننده همچنین در مرحله راه‌اندازی، فواصل بین گره‌ها را با ارسال بسته آزمایشی محاسبه و ذخیره می‌نماید. گره قرمزرنگ گره شروع‌کننده و محل قرارگیری ماشین مجازی هماهنگ‌کننده است و وظایفی ورودی ابتدا وارد این گره شده و سپس توسط زمان‌بند برای محاسبه به گره‌های دیگر سپرده می‌شوند. تنظیمات ارائه‌شده در جدول ۲ نشان داده شده است.

جدول ۲ تنظیمات ابر

پارامتر	توضیحات
گره	دو تا ده گره
ماشین مجازی	شش ماشین مجازی در هر گره با مشخصات یک گیگابایت حافظه، پردازنده تک‌هسته‌ای، فضای ذخیره‌سازی ۲۰۰ مگابایت
گراف جهت‌دار بدون دور	٦ گراف جهت‌دار بدون دور و روابط متفاوت
هماهنگ‌کننده	تعداد یک هماهنگ‌کننده در گره ۱

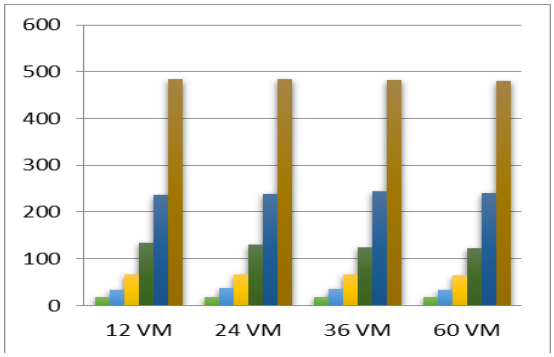
٦-۱ مقداردهی پارامترها

هر وظیفه ورودی پس از ورود به زمان‌بند ابتدا در حافظه نهان موردبررسی قرار می‌گیرد تا مشخص شود وظایفی مرتبط با آن در کدام گره‌ها اجراشده‌اند و بدین ترتیب relatednodes بدست می‌آید. همچنین آرایه relatednodes_priority تعداد وظایف مرتبط هر گره در آرایه relatednodes را مشخص می‌کند. سپس برای هر یک از گره‌های موجود در relatednodes، مقادیر energyusage, transfer, vmState به دست می‌آید. این مقادیر پس از مرتب‌سازی بر اساس اولویت‌ها آماده‌ی پردازش توسط فرآیند تحلیل سلسله مراتبی هستند.

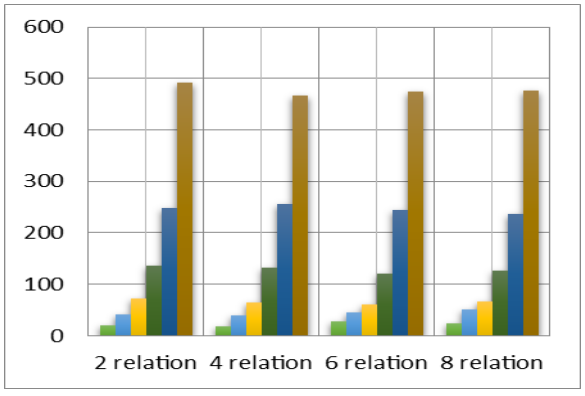
٦-۲ نتایج و مشاهدات

در قسمت پایانی پیاده‌سازی عملکرد الگوریتم زمان‌بندی بر اساس زمان اجرای هر وظیفه موردبررسی قرار گرفت. با تغییر در ماتریس اولویت‌ها در هر بار پیاده‌سازی الگوریتم زمان‌بندی، زمان اجرای هر وظیفه محاسبه شد تا مشخص گردد تأثیر کدام پارامتر بر کاهش انرژی و زمان اجرا بیشتر است. شکل (۲) زمان اجرای وظایف برای تمام اولویت‌ها را نشان می‌دهد.

این مقایسه نتیجه اجرای زمان‌بندی با اعمال تمامی پارامترها را نشان می‌دهد. نتایج بدست آمده نشان داد که افزایش تعداد ماشین‌های مجازی نمی‌تواند به‌طور مستقیم تغییر چشم‌گیری در زمان اجرای وظایف داشته باشد زیرا پارامتر VMState از راه‌اندازی‌های متعدد ماشین‌های مجازی جلوگیری می‌نماید. درواقع وقتی تعداد ماشین‌های مجازی از ۳٦ به ٦۰ افزایش یابد ممکن است از تمامی ماشین‌های مجازی جدید استفاده نشود زیرا راه‌اندازی ماشین مجازی جدید هم در مصرف انرژی و هم‌زمان اجرا تأثیر منفی خواهد داشت. با افزایش تعداد وظایف نیز روند نرمال رشد زمان اجرا را شاهد بودیم و بدین ترتیب افزایش بار تأثیر منفی در الگوریتم در برنداشت. شکل (۳) همین وضعیت را برای زمانی که تعداد ارتباطات بین وظایف افزایش می‌یابد نشان می‌دهد. با توجه به اینکه الگوریتم زمان‌بندی وظایف مرتبط را بدون انتقال و راه‌اندازی ماشین مجازی جدید اجرا می‌کند درنتیجه افزایش تعداد ارتباطات تأثیر مثبت بر زمان اجرای کل خواهد داشت.



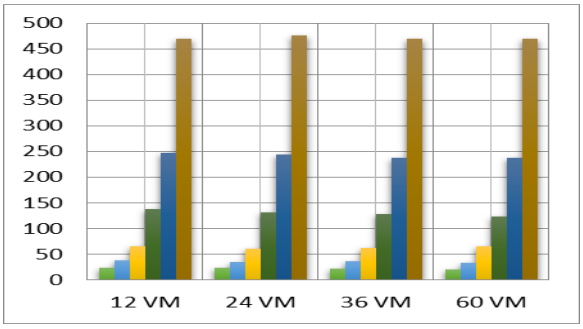
شکل ۲ نتایج با اعمال تمام پارامترها و تغییر تعداد وظیفه و ماشین مجازی



شکل ۳ نتایج با اعمال تمام پارامترها و تغییر تعداد

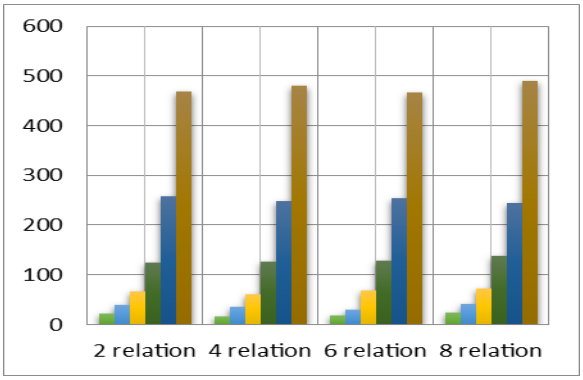
وظیفه و ارتباطات بین وظایف

جهت بررسی تأثیر هر یک از پارامترها بر نحوه اجرای الگوریتم زمان‌بندی، ابتدا پارامتر ارتباط بین وظایف یعنی Relti از مجموعه پارامترها حذف شد تا نتایج اجرای وظایف در این حالت مشاهده شود. نتایج نشان داد که زمان اجرای کل در مقایسه باحالت قبل کاهش خواهد داشت اما این روش مصرف انرژی بهینه را نمی‌تواند تأمین کند. با توجه به اینکه کاهش زمان اجرا بسیار ناچیز بوده لذا حذف این پارامتر الگوریتم زمان‌بندی را از رسیدن به اهداف مشخص‌شده دور خواهد کرد.



شکل ٤ نتایج با حذف پارامتر ارتباط بین وظایف و تغییر تعداد وظیفه و ماشین مجازی

این نتایج برای زمانی که تعداد وظایف و تعداد ماشین‌های مجازی متغیر است در شکل (٤) نشان داده شده است. برای حالت بعدی تعداد ارتباطات بین وظایف تغییر داده شده تا تأثیر حذف Relti بر اجرای الگوریتم مشاهده شود. نتایج بدست آمده برای این حالت در شکل (٥) نشان داده شده است.



شکل ١ نتایج با حذف پارامتر انرژی و تغییر تعداد وظیفه و ارتباطات بین وظایف

با توجه به روش خاص اتخاذشده در این مقاله و پارامترهای متفاوتی که در این تحقیق استفاده شده است تعداد مقالاتی که در سال‌های اخیر چاپ‌شده و قابلیت انطباق جهت مقایسه رادارند محدود است. گستردگی پارامترهای به‌کاربرده شده باعث تفاوت عمده در نوع پیاده‌سازی این الگوریتم زمان‌بندی‌شده است. این امر مقایسه دقیق و منطبق با سایر مطالعات و تحقیقات را دشوار نموده است. با توجه به مقایسه انجام‌شده با تحقیق [١٨] با اعمال پارامتر VMState به میزان ٣ درصد بهبود در کاهش مصرف انرژی حاصل شد. لازم به ذکر است تعداد پارامترهای در نظر گرفته‌شده با تعداد پارامترهای تحقیق [١٨] متفاوت است و این موضوع مقایسه کامل را دشوار کرده است. در مقایسه انجام‌شده با تحقیق [١٩] که سیاست کاهش انرژی را نیز موردبررسی قرار داده است دو پارامتر دما و انتشار CO_2 مقایسه گردید. نتایج بدست آمده بهبود میزان انتشار CO_2 را نشان داده است. درحالی‌که پارامتر دما در تحقیق حاضر موردبررسی قرار نگرفته است. بهبود حاصل‌شده در تحقیق [٢٠] نسبت به روش‌های EATSVM و ECTC سنجیده شده است. پس از انجام مقایسه‌های انرژی و زمان اجرا با توجه به اینکه تمامی پارامترها جهت مقایسه هم‌راستا نبوده‌اند، مصرف انرژی کاهش را نشان داده است. این بهبود ٥٪ بوده است. در مقایسه با تحقیق [٢١] تعداد وظایف مختلف موردبررسی قرار گرفت. در این نوع مقایسه پارامتر تعداد انتقالات و روابط بین وظایف بسیار تأثیرگذار بوده است، به‌گونه‌ای که بهبود زمان اجرا نسبت به نتایج تحقیق [٢١] قابل‌ملاحظه است. زیرا ما در اجرای وظایف با توجه به رابطه‌ای که وظایف در گراف DAG

دارند تعداد انتقالات و درنتیجه زمان اجرا را نیز کاهش می‌دهیم.

۷. نتیجه‌گیری

زمان اجرا، پارامتری است که ارتباط مستقیم با نحوه اجرای هر وظیفه دارد. یکی از موارد تأثیرگذار در اجرای وظایف و زمان کل اجرا، ارتباط بین وظایفی است به‌گونه‌ای که زمان بسیار زیادی از پردازش وظایف صرف رفت‌وآمد بین گره‌های مختلف می‌شود. برای مثال زمان‌بند تصمیم می‌گیرد t_1 در گره سوم اجرا شود، پس به ترتیب وظایف بعدی در گره‌های آزاد بعدی اجرا می‌شوند. مشکلی که این نوع از پردازش ایجاد می‌کند شامل انتقالات متعدد وظایف بین گره‌هاست. برای مثال t_2 که در گره ششم اجراشده نیاز به t_1 پیدا می‌کند و برای بدست آوردن آن نیاز به انتقال به گره سوم است. پارامتر Rel_{t_i} این انتقالات را پیش از وقوع و در مرحله زمان‌بندی کاهش می‌دهد.

به همین دلیل حذف این پارامتر از مجموعه پارامترهای زمان‌بندی تأثیر موقت بر زمان اجرای هر وظیفه دارد و زمان اجرای وظیفه به‌تنهایی بهبود می‌یابد، اما زمان اجرای کل به دلیل افزایش زمان موردنیاز برای انتقالات (که در زمان اجرای وظیفه دیده نمی‌شود) افزایش می‌یابد. لذا حذف پارامتر Rel_{t_i} اهداف زمان‌بندی را تأمین نمی‌کند. روش بد برای اجرای وظایف، بی‌اعتنایی به وضعیت ماشین‌های مجازی است به‌گونه‌ای که هر t_i با اولین ماشین مجازی خالی اجرا شود، در این صورت ماشین‌های مجازی به‌طور مرتبط خاموش خواهد شد که زمان زیادی صرف این بخش خواهد شد. لذا پارامتر VMState سعی دارد با مدیریت هوشمند ماشین‌های مجازی از این امر جلوگیری کند. از همین رو همان‌طور که در نتایج مشاهده شد دو نکته حائز اهمیت است یکی اینکه حذف پارامتر VMState از مجموعه پارامترها زمان اجرای هر وظیفه را بهبود می‌بخشد اما زمان اجرای کل را افزایش خواهد داد که خود باعث افزایش چشمگیر انرژی کل مصرفی می‌شود.

به‌عنوان کارهای آتی، اقدامات صورت گرفته در محیط واقعی و تحت‌فشار وظایف واقعی قرار خواهد گرفت، تا ایرادات احتمالی درروش محاسبه پارامترها و میزان تأثیرگذاری این پارامترها استخراج گردد. همچنین مدیریت ماشین‌های مجازی در بسترهای متفاوت نظیر VMWare و Virtual Box موردبررسی قرار خواهد گرفت.

مراجع

[1] Moganarangan, N., R. G. Babukarthik, S. Bhuvaneswari, MS Saleem Basha, and P. Dhavachelvan. "A novel algorithm for reducing energy-consumption in cloud computing environment: Web service computing approach." Journal of King Saud University-Computer and Information Sciences 28.1 (2016): 55-67.

[2] Yang, Song, Philipp Wieder, Ramin Yahyapour, and Xiaoming Fu. "Energy-aware provisioning in optical cloud networks." Computer Networks 118 (2017): 78-95.

[3] Dighe S, Vangal SR, Aseron P, Kumar S, Jacob T, Bowman KA, Howard J, Tschanz J, Erraguntla V, Borkar N, De VK. "Within-die variation-aware dynamic-voltage-frequency-scaling with optimal core allocation and thread hopping for the 80-core teraflops processor." IEEE Journal of Solid-State Circuits 46.1 (2010): 184-193.

[4] Ghanbari, Shamsollah, and Mohamed Othman. "A priority based job scheduling algorithm in cloud computing." Procedia Engineering 50.0 (2012): 778-785.

[5] Ghanbari, Shamsollah. "Priority-aware Job Scheduling Algorithm in Cloud Computing: A Multi-criteria Approach." Azerbaijan Journal of High Performance Computing, Vol 2, Issue 1, 2019, pp.29-38.
[6] Kazemi, Mojtaba, Shamsollah Ghanbari, and Manochehr Kazemi. "Divisible Load Framework and Close Form for Scheduling in Fog Computing Systems." In International Conference on Soft Computing and Data Mining, pp. 323-333. Springer, Cham, 2020.
[7] Rong, Huigui, Haomin Zhang, Sheng Xiao, Canbing Li, and Chunhua Hu. "Optimizing energy consumption for data centers." Renewable and Sustainable Energy Reviews 58 (2016): 674-691.
[8] Aghababaeipour, Ali, and Shamsollah Ghanbari. "A new adaptive energy-aware job scheduling in cloud computing." In International Conference on Soft Computing and Data Mining, pp. 308-317. Springer, Cham, 2018.
[9] Ding, Youwei, Xiaolin Qin, Liang Liu, and Taochun Wang. "Energy efficient scheduling of virtual machines in cloud with deadline constraint." Future Generation Computer Systems 50 (2015): 62-74.
[10] Dong, Ziqian, Ning Liu, and Roberto Rojas-Cessa. "Greedy scheduling of tasks with time constraints for energy-efficient cloud-computing data centers." Journal of Cloud Computing4.1 (2015): 5.
[11] Hosseinimotlagh, Seyedmehdi, Farshad Khunjush, and Rashidaldin Samadzadeh. "Seats: smart energy-aware task scheduling in real-time cloud computing." The Journal of Supercomputing 71.1 (2015): 45-66.
[12] Xiong, Ying Yidu, and Yan Yan Wu. "Cloud computing resource schedule strategy based on pso algorithm." Applied Mechanics and Materials. Vol. 513. Trans Tech Publications, 2014.
[13] Ghribi, Chaima, Makhlouf Hadji, and Djamal Zeghlache. "Energy efficient vm scheduling for cloud data centers: Exact allocation and migration algorithms." 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing. IEEE, 2013.
[14] Beloglazov, Anton, Jemal Abawajy, and Rajkumar Buyya. "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing." Future generation computer systems 28.5 (2012): 755-768.
[15] Chen, Shi, Jie Wu, and Zhihui Lu. "A cloud computing resource scheduling policy based on genetic algorithm with multiple fitness." 2012 IEEE 12th International Conference on Computer and Information Technology. IEEE, 2012.
[16] Luo, Liang, Wenjun Wu, Dichen Di, Fei Zhang, Yizhou Yan, and Yaokuan Mao. "A resource scheduling algorithm of cloud computing based on energy efficient optimization methods." 2012 International Green Computing Conference (IGCC). IEEE, 2012.
[17] Van Do, Tien, and Csaba Rotter. "Comparison of scheduling schemes for on-demand IaaS requests." Journal of Systems and Software 85.6 (2012): 1400-1408.
[18] Juarez, Fredy, Jorge Ejarque, and Rosa M. Badia. "Dynamic energy-aware scheduling for parallel task-based application in cloud computing." Future Generation Computer Systems 78 (2018): 257-271.
[19] Adhikari, Mainak, and Satish Narayana Srirama. "Multi-objective accelerated particle swarm optimization with a container-based scheduling for Internet-of-Things in cloud environment." Journal of Network and Computer Applications137 (2019): 35-61.
[20] Ismail, Leila, and Huned Materwala. "EATSVM: Energy-Aware Task Scheduling on Cloud Virtual Machines." Procedia Computer Science 135 (2018): 248-258.
[21] Ch, Sudhakar, and T. Ramesh. "Energy efficient VM scheduling and routing in multi-tenant cloud data center." Sustainable Computing: Informatics and Systems 22 (2019): 139-151.