تاریخ دریافت مقاله: ۹۶/۰۱/۲۹
تاریخ پذیرش مقاله: ۹۶/۰۴/۱۰

ارائه روشی جهت بهبود انرژی در مراکز داده ابری با مهاجرت خدمت بر روی کارسازهای همکار

محمد رستمی*
عضو باشگاه پژوهشگران جوان دانشگاه آزاد اسلامی واحد دهاقان، اصفهان، ایران
پست الکترونیکی: mohammadrostami@dehaghan.ac.ir

سید سعید آیت
دانشیار گروه مهندسی کامپیوتر و فناوری اطلاعات دانشگاه پیام نور
پست الکترونیکی: dr.ayat@pnu.ac.ir

الهه نشاط
کارشناس ارشد نرم افزار کامپیوتر دانشگاه آزاد اسلامی واحد تهران شمال، تهران
پست الکترونیکی: Elahe_Neshat@yahoo.com

چکیده

محاسبات ابری به برنامه و خدمت‌هایی اشاره دارد که در یک شبکه توزیع شده اجرا می‌شود و از منابع مجازی استفاده می‌کند، همچنین از طریق پروتکل‌های رایج اینترنت و استانداردهای شبکه قابل دستیابی می‌باشند. تقاضاهای رو به رشد، زیرساخت‌های ابری مصرف انرژی را در مراکزداده‌ها افزایش داده است که این موضوع مهمی است. مصرف بالای انرژی نه تنها منجر به هزینه‌های عملکردی بالا می‌شود و سوددهی فراهم‌کنندگان ابر را کاهش می‌دهد بلکه به انتشار بالای کربن منجر می‌شود که برای محیط زیست مناسب نیست. تحقیقات زیادی پیشنهاد شد تا مراکز داده‌هایی با کارایی انرژی بالا ایجاد گردد و از تکنیک‌هایی مثل مجازی‌سازی و تثبیت استفاده کنند. این راه‌حل‌ها مقرون به صرفه هستند امّا مستقیما اثرات محیطی انتشارات CO2 را بیان نمی‌کنند.

در روش پیشنهادی به تجمیع خدمت‌های همکار پرداخته شده است. طی عمل مهاجرت خدمت‌ها از یک میزبان که انرژی مصرفی آن بیش از حد یا کمتر از حد می‌باشد به میزبان دیگر که حاوی خدمت‌های همکار است منجر به کاهش مصرف انرژی مرکز داده می‌شود. روش پیشنهادی با سه الگوریتم که از نظر مهاجرت زنده بهترین الگوریتم‌های موجود و مناسب‌تر با موضوع پژوهش می‌باشند، مقایسه شد. در نتیجه روش پیشنهادی در حدود ۱۱/۶ درصد در صرفه جویی مصرف انرژی مرکز داده بهبود یافته است.

کلید واژه‌ها: محاسبات ابری، محاسبات سبز، کاهش مصرف انرژی، مهاجرت خدمت‌ها.

۱- مقدمه

در سال‌های اخیر توسعه دهندگان خدمات فناوری اطلاعات، مثل آی‌بی‌ام، مایکروسافت، گوگل و سایر سازمان‌های بزرگ مشابه، مراکز داده را برای خدمات

محاسبات ابری توسعه داده‌اند. همچنین مراکز داده دارای هزاران خدمت دهنده و سوئیچ می‌باشد. در مراکز داده بزرگ تبادل پیام میان خدمت‌هایی که با هم ارتباطات زیادی دارند ولی در کارسازهای مختلفی نگهداری می‌شوند سبب افزایش هزینه ارتباطی و تأخیر در پاسخ می‌شود. پژوهش‌های زیادی نشان داده‌اند که ارتباطات خدمت‌ها در بین کارسازهای مختلف نه‌تنها باعث هدر رفتن منابع ارتباطی می‌شوند، بلکه باعث مصرف بیهوده منابع دیگر مانند پردازنده می‌شوند[۳,۶,۷,۱۸] . از طرف دیگر به دلیل گرمای تولید شده توسط این مراکز داده، نیاز به تجهیزات خنک کننده می‌باشد. در حقیقت مراکز داده در محاسبات ابری مقدار زیادی از انرژی الکتریکی را مصرف می‌کنند و این به نوبه خود باعث افزایش هزینه‌های عملیاتی و تولید دی اکسید کربن در محیط پیرامون می‌شود.

در این پژوهش تأکید بیشتر در این است که افزایش محلی‌سازی خدمت‌ها با کنترل مهاجرت خدمت‌ها انجام شود. در واقع به موضوع مهاجرت خدمت‌های همکار با هدف کاهش مصرف انرژی پرداخته شده است. نتایج این تحقیق باعث افزایش محلی‌سازی ارتباطات خدمت‌ها و به دنبال آن کاهش مصرف انرژی مراکز داده‌ها می‌شود. برای مراکز داده، کاهش هزینه‌های ارتباطی میان خدمت‌ها اهمیت و ضرورت زیادی دارد که این موضوع ارتباط مستقیمی با محل استقرار خدمت‌های مرتبط در مرکز داده دارد و استقرار صحیح خدمت‌ها در مرکز داده می‌تواند سبب کاهش مصرف انرژی مرکز داده شود. به طور کلی دو نوع مهاجرت زنده و غیر زنده وجود دارد که در روش پیشنهادی از مهاجرت زنده استفاده شد، یعنی ماشین مجازی در حالی که روشن است و به درخواست‌های رسیده از سوی کاربران پاسخ می‌دهد به ماشین مقصد منتقل می‌شود، بنابراین سیستم دچار وقفه نمی‌گردد. برای محلی‌سازی خدمت‌ها، با استفاده از یک الگوریتم، خدمت‌هایی که با یکدیگر همکاری زیادی دارند به یک میزبان منتقل می‌شوند، در مراکز داده بزرگ خدمت‌های مختلفی وجود دارد که در کارسازهای مختلفی نگهداری می‌شوند، ارتباطات خدمت‌ها بین کارسازهای مختلف باعث هدر رفتن منابع و افزایش مصرف انرژی می‌شود، بنابراین چنانچه بتوانیم خدمت‌هایی که با هم ارتباط زیادی دارند که خدمت همکار خوانده می‌شوند، در یک کارساز قرار دهیم، هزینه ارتباطی صفر می‌شود و در مصرف انرژی نیز صرفه‌جویی می‌گردد، به این عمل انتقال، مهاجرت گفته می‌شود.

میزبانی که درحال مصرف انرژی بیش از حد است و یا میزبانی که مصرف انرژی آن به قدری کم است که بهتر است خاموش شود، شناخته می‌گردد. در این زمان الگوریتم پیشنهادی برای مهاجرت اجرا می‌شود. هدف، انتقال خدمت‌ها از این میزبان به دیگری با استفاده از مهاجرت زنده می‌باشد به طوری که قرارداد و نیازهای کاربر نقض نگردد. مهاجرت زنده این امکان را می‌دهد که بعد از مهاجرت، سیستم بدون شروع از ابتدا، از همان جا که دچار وقفه شد کار خود را از سر بگیرد. نشان داده شده است که زمان از کارافتادگی سیستم به دلیل این نوع مهاجرت بسیار کم است (یعنی ۶ میلی ثانیه) به همین دلیل افت عملکرد چندانی مشاهده نمی‌شود. خدمت‌هایی که انتخاب شدند به میزبان جدیدی که حاوی خدمت‌های همکار است، برای ادامه اجرای خود نگاشت می‌شوند، این نگاشت باعث می‌شود هزینه ارتباطی بین خدمت‌های همکار که همان خدمت‌هایی بودند که مدام نیاز داشتند برای تبادل پیام باهم ارتباط داشته باشند صفر شود و در نتیجه این فرآیند منجر به کاهش مصرف انرژی مرکز داده می‌شود.

الگوریتم پیشنهادی تفاوت‌های عمده با الگوریتم‌های دیگر خواهد داشت که برخی از این تفاوت‌ها عبارتند از:

- ارائه روشی برای تجمیع خدمت‌های همکار
- کاهش مصرف انرژی با هدف همگون‌سازی گردش کار
- مهاجرت خدمت‌ها با رویکرد کاهش مصرف انرژی
- ارائه راهکاری برای مهاجرت خدمت‌ها بر اساس یک روش اولویت‌بندی

شبیه‌سازی و آزمایش توسط کد منبع کلودسیم انجام شده است. در ارزیابی‌ها مشاهده می‌گردد روش پیشنهادی حدود ۱۱/۶ درصد نسبت به روش‌های پیشین بهبود دارد و در نتیجه روش پیشنهادی در حدود ۱۱/۶ درصد در مصرف انرژی مرکز داده صرفه‌جویی می‌نماید.

ساختار ادامه مقاله به این صورت می‌باشد: در بخش دوم کارهای مرتبط مورد بررسی قرار می‌گیرد. در بخش سوم روش پیشنهادی تشریح می‌شود. بخش چهارم به ارزیابی و تفسیر نتایج اختصاص دارد. در نهایت بخش پنجم نیز به نتیجه‌گیری و کارهای آینده می‌پردازد.

۲- کارهای مرتبط

رشد سریع تقاضا برای استفاده از منابع محاسباتی ابری توسط مراکز داده عظیم، موجب مصرف مقادیر بالای انرژی شده است. یکی از راهکارهای کاهش مصرف انرژی در سیستم‌های پردازش ابری استفاده از الگوریتم‌های تجمیع خدمت‌های همکار است. این الگوریتم‌ها با مهاجرت دادن خدمت‌ها براساس اولویت، از میزبان‌هایی که بار کاری خیلی زیاد و یا خیلی کم دارند (به گونه‌ای که موجب مصرف انرژی زیادی می‌شوند) به میزبان‌هایی که حاوی خدمت‌های همکار هستند، باعث استفاده بیشتر از منابع موجود و کاهش مصرف انرژی می‌شود. روش پیشنهادی را با سه سیاست، حداقل زمان مهاجرت (MMT)، سیاست انتخاب تصادفی(RC)، حداقل مصرف (استفاده) (MU) مقایسه کرده که در طی آن پارامترهای کاهش مصرف انرژی، تعداد مهاجرت‌ها، تعداد میزبان‌هایی که خاموش شده‌اند، مورد مقایسه و بررسی قرار گرفته است. برای پیاده‌سازی و ارزیابی الگوریتم از شبیه‌ساز معروف کلودسیم استفاده شد. که به بهبود قابل توجهی دست پیدا شد.

در ادامه به بررسی روش‌های پیشین، مزایا و معایب پرداخته می‌شود.

ایکسونگ فو و همکارش یک سیاست جایگذاری ماشین مجازی ارائه نمودند به طوری که ماشین مجازی که باید جابجا شود را در یک میزبان که کمترین ضریب همبستگی را دارد، قرار می‌دهد. همچنین علاوه بر در نظر گرفتن بهره‌وری پردازنده، یک سری متغیر جدید را که بر میزان رضایت منابع تاثیر می‌گذارد تعریف نمودند. عیب این روش این است که قرار دادن ماشین مجازی در میزبان با کمترین ضریب همبستگی، مصرف انرژی را افزایش می‌دهد[۱].

وینوتینا و همکارانش به بررسی راهبردهای مختلف تخصیص منابع در رایانش ابری پرداخته و مزایا و معایب این روش را بررسی کرده و چالش‌های پیش روی این حوزه از جمله کاهش مصرف انرژی را نیز بیان نمودند [۲].

آقای پهلوان و همکارانش ساختار شبکه و سرمایشی مرکز داده را در زمانی که فکر تجمیع ماشین‌های مجازی بود، در نظر گرفتند تا بدون به خطر انداختن توافق نامه کیفیت خدمت مداربند و مسیریاب کمتری به‌کار گرفته شود. بدین ترتیب، در مزیت این روش تجهیزات مسیریابی و سرمایشی خاموش شده تا کاهش مصرف انرژی داشته باشیم [۳].

راجا احمدا و همکارانش به بررسی شمای پهنای باند، تجمیع کارساز، بهینه‌سازی با DVFS و روش‌های بهینه‌سازی منابع ذخیره‌سازی بر روی شبکه‌های گسترده، معماری‌های مختلف تجمیع ماشین‌های مجازی بر روی کارسازهای کمتر و همچنین عوامل تاثیرگذار در این معماری پرداخته‌اند. مزیت این روش بررسی شمای پهنای باند و تجمیع ماشین‌های مجازی بر روی کارسازهای کمتر ایده مناسبی جهت کاهش انرژی بود [۴].

جیپنگ جاوو و همکارانش از الگوریتمی بر مبنای تئوری بهینه‌سازی ازدحام ذرات برای جایگذاری ماشین مجازی در یک مرکز داده استفاده کردند. این الگوریتم به کاهش تعداد مهاجرت‌های ماشین مجازی و بهره‌وری منابع بهبود بخشید. مزیت این روش این است که توانست با استفاده از یک الگوریتم تعداد مهاجرت‌های ماشین مجازی را کاهش

دهـد و در نتیجه بهره وری منابع بـالا رود. از آنجایی که میزبان مناسـبی انتخاب نشـده بود همچنان در مصرف انرژی هدر رفت، داشت [۵].

وان و همکارانش روشی معرفی نمودند که ماشین‌های مجازی بر اسـاس این‌که کدامیک مکمل یکدیگر هستند، در طی چند تکرار در گروه‌بنـدی مختلف قرار می‌گیرند. ایده خوب این روش به عنوان مزیت، در نظر گرفتن ماشین‌های مجازی مکمـل هم بود کـه در این صورت ماشـین‌های مجازی که باهم مکمل هسـتند در یک گروه قرار می‌گیرند و در نتیجـه به علت عدم پراکندگی انرژی کمتری مصرف می‌نمایند [۶].

یونگ و همکارانش با استفاده از تجمیع کل ماشین‌های مجازی موجود در یک سیستم ابری بر روی یک کارساز، تا حد ممکن از روشـن ماندن کارسازهای دیگر جلوگیری می‌نمودند، بدین ترتیب میزبان‌های بی‌استفاده را به حالت خاموش می‌بردند. با این تکنیک کارسـاز کمتری روشـن می‌باشـد و پیرو آن انرژی مصرفی کارساز و مرکز داده کاهش پیدا می‌کنـد. مزیت این روش تجمیع ماشـین‌های مجازی و خاموش کردن میزبان‌های بی‌استفاده می‌باشد. و عیب روش به‌کار رفته برای تجمیع ماشـین‌های مجازی زمان‌بربودن آن بود [۷].

ناگپـور و همکارانـش یک الگوریتـم تخصیص منابع خـودکار در رایانش ابری به نحوی که منابع را در اختیار کاربران قرار دهد، پیشـنهاد دادند. این الگوریتم به وسیله تعادل بار و پیش‌بینی حجم کاری آینده از سربار بارکاری جلوگیـری و به بهبـود کارایـی کمک نمود. بـه عبارتی بهره‌وری منابع با کمترین مصرف انرژی. عیب این روش، پیش‌بینی حجم کاری آینده بود که زمان‌بر است [۸].

ژنگ و همکارانش یک روش جایگذاری پویای مجازی جدیـد در رایانش ابری ارائـه دادند. با ایـن روش تعداد گره‌هـای کمتری فعال اسـت تا انرژی کلـی در یک مرکز داده را کاهش دهد. همچنیـن این روش جایگذاری بر پایه ریاضیات بنا شده اسـت. مزیت این روش این است که از فاکتورهـای مهم دیگری که در مصـرف انرژی تاثیرگذار اسـت، اسـتفاده می‌کند که تا حد ممکن انرژی مصرفی را کاهش دهد[۹].

ساتورو یک الگوریتم جایگذاری سریع ماشین مجازی برای مـدل تقاضای پویا ارائه داد. براسـاس گزارش‌های گرفته شده از خروجی آن متوجه شدند یک داد و ستد بین تعداد ماشین‌های فیزیکی و تعداد مهاجرت‌ها وجود دارد. از معایب این روش عبارتند از: هزینه‌بر بودن و این که انرژی مصرفی را چندان کاهش نمی‌دهد [۱۰].

عاطفه خسـروی و همکارانش یک الگوریتم جایگذاری ماشـین مجازی برای افزایش سازگاری محیط با در نظر گرفتـن مراکـز داده توزیع شـده با نرخ اثـرات کربنی و مصرف انرژی مختلف ارائه نمودند[۱۱].

بیلوگلازوو و همکارانش، الگوریتمی برای کارسـازی که بهترین مصرف انرژی را داشـته باشـد، ارائه نمودند. با مرتب‌سـازی ماشـین‌های مجازی به ترتیـب نزولی و بـر مبنای میزان بهره‌بـرداری از پردازنده، تخصیصی را برگزیدند که طی آن کمترین افزایش توان به وقوع بپیوندد که مزیت این روش طبق ارزیابی، به استفاده بهینه از میزان مصرف انرژی و منابع در دسترس منجر شده است[۱۲].

مارتیـن بیچلـر و همکارانش بـا اسـتفاده از روش برنامه‌ریزی خطی طرحی برای تخصیص ماشین مجازی پیاده نمودند. اگرچه از متغیرهای دودویی استفاده کردند اما عیب این روش فقط برای تخصیص منبع ایستا جوابگو بود و ماهیت پویای محیط ابری را پوشش نمی‌دهد[۱۳].

بابـروف و همکارانش یک الگوریتم بـرای یکپارچگی ماشین‌های فیزیکی که به صورت دوره‌ای برای به حداقل رساندن ماشـین‌های روشن مورد استفاده قرار می‌گرفت ارائـه نمودند که مزیـت آن کاهش مصرف انـرژی با به حداقل رساندن تعداد ماشین‌های روشن بود[۱۴].

اسپیتکامپ و همکارانش یک فرمول خطی برای مسئلهٔ یکپارچگی پویای ماشین‌های مجازی ارائه کردند، و نگاشت برخی از ماشین‌های مجازی را به کارسازهای بخصوصی

که ویژگی‌های خاصی دارند منحصر کردند. همچنین در فرمول سازی خود به معیار هزینه نیز توجه داشتند. مزیت این روش این است که معیار هزینه را در نظر گرفته ولی عیب آن این‌که به بهینه کردن مصرف انرژی خیلی توجهی نداشت [۱۵].

از دیگر کارهایی که در این زمینه انجام شده است، پژوهشی است که شی و هانگ ارائه داده‌اند[۱۶]. دو نوع سطح عملکرد برای خدمت‌ها متصور شدند، سخت و نرم. در نوع سخت اگر خدمتی که ارائه می‌شود دقیقا در سطح درخواست شده در قرارداد نباشد آنگاه هیچ درآمدی به خدمت دهنده نمی‌رسد. اما در نوع نرم مشتری‌ها می‌توانند سطوح عملکرد ضعیف‌تر را نیز تحمل کنند. اما درآمد به‌دست آمده طبق جدول قیمت‌گذاری با کاهش روبه‌رو می‌شود. از مزایای این روش این است که سطح درخواست خدمت کاربر را نیز در نظر داشت و روش خود را بدون در نظر گرفتن و با در نظر گرفتن سطح درخواست خدمت کاربر و برآورد هزینه بررسی نموده است. برای حل مسئلهٔ تخصیص منبع علاوه بر عامل افزایش توان که در الگوریتم ارائه شده توسط بویا در نظر گرفته شده بود[۱۷]، متغیر سود را هم در نظر گرفتند. یعنی با هر تخصیص ماشین مجازی به میزبان، سود حاصله نیز محاسبه شد. از مزایای این روش در نظر گرفتن سود حاصله بود که یکی از معیارهای مهم محسوب می‌گردد.

به‌طور کلی با مطالعه روش‌های فوق نتیجه گرفته می‌شود که چنانچه ماشین‌های مجازی به گونه‌ای مهاجرت داده شود که ماشین‌های مجازی مکمل یا همکار در میزبان مناسب در کنار هم تجمیع شوند، مصرف انرژی کاهش می‌یابد. بنابراین در ایده پیشنهادی به انتخاب ماشین مجازی مناسب، تجمیع ماشینهای مجازی و انتخاب میزبان مناسب در جهت کاهش مصرف انرژی پرداخته می‌شود.

۳- روش پیشنهادی

در حالت کلی مسئلهٔ تخصیص خدمت‌ها به میزبان‌ها را می‌توان به سه زیر مسئله تقسیم نمود. زیرمسئلهٔ اول این است که چه زمانی باید خدمت مهاجرت کند؟ زیرمسئلهٔ دوم به این‌که کدام خدمت مهاجرت کند. زیرمسئلهٔ سوم به پرسش خدمت به کجا مهاجرت کند باید پاسخ دهد. این‌که چه زمانی باید خدمت‌های یک میزبان مهاجرت کنند، زمانی که انرژی مصرفی یک میزبان در مرکز داده بیشتر از حد آستانه[1] باشد الگوریتم اجرا و خدمت‌های آن میزبان شروع به مهاجرت می‌کنند.

تمرکز اصلی این تحقیق بر روی حل زیرمسئلهٔ دوم و سوم می‌باشد. برای حل زیرمسئلهٔ دوم، ابتدا خدمت‌هایی که مدام با یکدیگر در ارتباط و خدمت‌های همکار خوانده می‌شوند، یافته می‌شود، سپس خدمتی که کمترین همکاری را با بقیه خدمت‌های آن میزبان دارد، برای مهاجرت انتخاب می‌شود.

برای حل زیرمسئلهٔ سوم، برای نگاشت خدمت به یک میزبان، به میزبانی نگاشت می‌شود که حاوی خدمت‌های همکار آن خدمت باشد و بدین‌گونه هزینه ارتباطی صفر شود که در نتیجه انرژی مصرفی کمتر می‌گردد.

بنابراین فرآیند مهاجرت در الگوریتم ارائه شده در سه مرحله اصلی انجام می‌شود:

- پیدا کردن خدمت‌های همکار ServicePartner(v)
- انتخاب خدمت مناسب برای مهاجرت Select Service ()
- انتخاب مقصد مناسب برای مهاجرت خدمت Select Host(v, v')

۳-۱ مرحله اول : پیدا کردن خدمت‌های همکار

خدمت‌هایی که باهم ارتباطات زیادی دارند که همان خدمت‌های همکار هستند، ولی در کارسازهای مختلفی نگهداری می‌شوند سبب افزایش هزینه ارتباطی می‌شوند. هنگامی که خدمت‌های همکار کنار هم قرار می‌گیرند، هزینه ارتباطی صفر می‌شود، منابع کمتری مصرف می‌شود و در نتیجه هم زمان پاسخ و هم انرژی مصرفی

1- Threshold

کاهش می‌یابد. جهت رسیدن به این هدف خدمت‌ها را به میزبانی که حاوی خدمت‌های همکار هستند انتقال می‌دهیم. بنابراین در این مرحله خدمت‌های همکار هر خدمت یافت می‌شوند.

برای به‌دست آوردن خدمت‌های همکار یک خدمت، باید مجموع تعداد دفعاتی که خدمت با هر خدمت ملاقات داشته و همین‌طور مجموع هزینه‌های ارتباطی خدمت با خدمت‌هایی که در ارتباط است محاسبه شود. ابتدا مجموع هزینه ارتباطی هر خدمت با خدمت‌های دیگر و مجموع تعداد دفعاتی که هر خدمت با خدمت‌های دیگر ملاقات داشته را محاسبه و هر یک از پارامترهای فوق را با استفاده از رابطه (۱) نرمال می‌شود، سپس با استفاده از رابطه (۲) رتبه همکاری هر خدمت با خدمت‌های دیگر به‌دست می‌آید و با استفاده از رابطه (۳) مجموع رتبه همکاری هر خدمت با تمام خدمت‌ها محاسبه می‌شود.

$$\text{Norm}(x_i)=\frac{x_i-\min(x)}{\max(x)-\min(x)}$$ رابطه (۱)

مجموعه x به صورت $\{x_1,x_2,\ldots,x_n\}$ می‌باشد و می‌تواند مجموعه‌ای از هزینه‌های ارتباطی خدمت‌ها با خدمت x_i یا تعداد ملاقات‌های هر خدمت با خدمت x_i باشد.

min(x) کوچک‌ترین مقدار مجموعه x .

max(x) بزرگ‌ترین مقدار مجموعه x .

رابطه (۲)

$$\text{RankCooperator}(V_i , V_j) = Norm\left(sumCommunicationCost(V_i , V_j)\right) + Norm(sumMeet(V_i , V_j))$$

رابطه (۳)

$$\text{SumRankCooperator}(V_i) = \sum_{j=1}^{m} \text{RankCooperator}(V_i , V_j)$$

Where m is a Count Number of Services

روش پیشنهادی در سه مرحله متوالی اجرا می‌شود. ورودی این الگوریتم لیستی از میزبان‌ها و لیستی از خدمت‌ها می‌باشد. در ابتدا لیستی از DAG[۲]ها (گراف جهت‌دار بدون دور) با همه مشخصات خدمت‌ها، فرزندان و هزینه ارتباطی بین گره‌ها به عنوان ورودی الگوریتم مشخص می‌شود. سپس همه خدمت‌ها در یک لیست V قرار داده می‌شود و همه میزبان‌ها در یک لیست H قرار می‌گیرند و میزبان X که قرار است خدمت‌های آن مهاجرت کنند مشخص است. در واقع خدمت‌های درخواستی به صورت DAGهایی دیده می‌شوند و می‌توان با استفاده از پارامترهای استخراج شده از DAGها اولویت هر خدمت را مشخص نمود تا هنگامی که قرار بر مهاجرت یک خدمت بود، خدمت با اولویت پایین را مهاجرت داد. پارامترهای ورودی در زیر نمایش داده شده است.

Input
1 Input List of DAG(Full) 2 Input List of Execute Time on every Service 3 List (*V*)=All Service 4 List(*H*)=All Host 5 Host *X*

خروجی: نگاشت مناسب خدمت‌های با اولویت در میزبان مناسب

Output
A mapping of Servers to Host

در تابع Main، با استفاده از تابع () computing Rank، رتبه هر خدمت که در کارساز X وجود دارد به‌دست می‌آید. سپس با استفاده از تابع Computing MinEST(vi) و Computing MaxEFT(vi)، MinEST خدمت vi و MaxEFT خدمت Vi به‌دست می‌آید. خدمت Vi به (CandidatedList) لیست خدمت‌هایی که قرار است مهاجرت پیدا کنند وارد می‌شود. پس از آن‌که برای تک تک خدمت‌ها این سه پارامتر (MinEST, MaxEFT ,Rank) محاسبه شد و هر خدمت به لیست (CandidatedList) اضافه شد، لیست به صورت نزولی براساس مقدار رتبه (Rank)، مرتب می‌شود و اولین مورد آن که خدمت با اولویت بالاتر است انتخاب می‌شود (خدمت V). سپس این خدمت به تابع ServicePartner (v) فرستاده می‌شود تا خدمت همکار، خدمت V، (خدمتی که بیشترین ملاقات را با خدمت V داشته است) مشخص شود، خروجی این تابع، نام خدمت همکار یعنی خدمت v' و تعداد دفعاتی است که خدمت V، خدمت همکار v' را صدا زده

2- Direct Acyclic Graph

است که مقدار آن در متغیر m قرار می‌گیرد. اکنون این سه متغیر، یعنی V (خدمتی که باید به میزبانی نگاشت شود)، v' (خدمت همکار خدمت V) و m (تعداد دفعاتی که خدمت V، خدمت v' را صدا زده است) به تابع SelectHOST(v, v',m) داده می‌شود تا میزبان مناسب برای خدمت V را پیدا کند و این خدمت را به آن میزبان اختصاص دهد. در نهایت خدمت V از (CandidatedList) حذف می‌شود.

Input: List of DAG(Full), List of Execute Time on every Service, ServiceList,HostList
Output: A mapping of Servers to Host
Main(v, v',m)

```
1  {
2    foreach serviceType (Vi) in HOST X
3      {
4        Computing Rank(vi)
5        Computing MinEST(vi)
6        Computing MaxEFT(vi)
7        Insert vi to CandidatedList
8      }
9    sort CandidatedList
10       Select first item (v) from CandidatedList
11       [v']=ServicePartner (v)
12       SelectHOST(v, v') to Migration Service v
13   remove v from CondidatedList
14 }
```

شبه‌کد تابع computing Rank که در این تابع برای محاسبه اولویت هر خدمت باید مجموع سطح، مجموع تعداد فرزندان و مجموع هزینه ارتباطی هر خدمت در تمام DAGها به‌دست آورده شود، سپس برای به‌دست آوردن رتبه هر خدمت مجموع متوسط هر یک از این سه پارامتر محاسبه می‌شود. قابل ذکر است که هرکدام از این پارامترها ابتدا بهنجار می‌شود و سپس در مجموع محاسبه می‌شود.

SumLevel (vi) مجموع سطح خدمت vi در هر DAG است.

SumNumber of children (vi) مجموع تعداد فرزندان خدمت vi در هر DAG است.

SumCommunication cost (vi) مجموع هزینه ارتباطی خدمت vi در هر DAG است.

N تعداد DAGها است.

Input: service vi
Output: Rank(vi)
Computing Rank (service vi)

```
1  {
2    foreach DAG
3      {
4        SumLevel(vi) +=NormLevel ( vi ) in DAG j
5        SumNumber of children (vi) +=NormCountOfChildren (vi) in DAG j
6        SumCommunication cost (vi) +=NormComputtingCommunicationCost(vi) in DAG j
7      }
8    N=The Number of DAG
9    Rank(vi)=SumNumber of children (vi)/N - SumLevel(vi)/N + SumCommunication cost (vi )/N
10   return (Rank(vi) )
11 }
```

برای به‌دست آوردن MinEST، ابتدا برای هر DAG مقدار EST خدمت vi را محاسبه نموده و در نهایت از بین مقادیر به‌دست آمده کوچک‌ترین آن را به عنوان MinEST(vi) نشان می‌دهد.

Input: service vi
Output: MinEST(vi)
Computing MinEST (vi)

```
1  {
2    foreach DAG
3      {
4        if (i==1) then EST(Vi)=0
5        else
6          {
7            EST(Vi)= EST(Vh) + ExcuteTime(Vh) + MAXCommunicationCost(Vh,Vi)
8          }
9      }
10   return (Min EST(Vi))
11 }
```

برای به‌دست آوردن MaxEFT، ابتدا برای هر DAG مقدار EFT خدمت vi را محاسبه نموده و در نهایت از بین مقادیر به‌دست آمده بزرگ‌ترین آن را به عنوان MaxEFT(vi) نشان داده می‌شود.

Input: service vi
Output: MaxEFT(vi)
Computing MaxEFT(vi)

```
1  {
2    foreach DAG
3      {
4        EFT(Vi)=EST(Vi)+ ExcuteTime(Vi)
5      }
6    return (Max EFT(Vi))
7  }
```

۳-۲ مرحلـــه دوم : انتخاب خدمت مناســـب برای مهاجرت

زمانی که میزبانی بیش از حد آســـتانه، انرژی و یا کمتر از حد آستانه انرژی مصرف می‌کند به گونه‌ای که بهتر است خاموش شود، نیاز است که مهاجرت انجام شود. برای انتخاب خدمت مورد نظر برای مهاجرت، از بین خدمت‌های آن میزبان خدمتی مهاجرت داده می‌شـــود که با اســـتفاده از رابطه (۳) کمترین همکاری را با بقیه خدمت‌های آن میزبان دارد.

جهت یافتن خدمت همکار باید برای تمام خدمت‌های j در تک تک DAGها بررسی شود که خدمت v چندبار با خدمت j ارتباط داشـــته اســـت و مجموع هزینه‌های این ارتباط در تمام DAGها چقدر بوده است. تعداد دفعات ملاقات و مجموع هزینه‌های ارتباطی خدمت j در یک لیست قرار داده می‌شود و هنگامـــی که این دو پارامتر برای همه خدمت‌ها بررســـی شد، لیست بر اساس حداکثر هزینه ارتباطی مرتب می‌شود و چنانچه خدمت‌هایی وجود داشتند که هزینه‌های ارتباطی هریک با خدمت v یکسان بود، آنگاه براساس حداکثر تعداد دفعات ملاقات با خدمت v، مرتب سازی انجام می‌شود. اولین مورد از این لیست به عنوان خدمت همکار v انتخاب می‌شود.

Input: service vi
Output: service v // ServicePartner (service v)
ServicePartner (service v)

```
1 }
2    foreach service j
3      {
4          communication cost(service v,service j)=0;
5          Meet (service v,service j) =0;
6           RankCooparator (service v,service j)=0;
7      }
8     foreach DAG  i
9       {
10          if service v and service j in workflow i  have
communication
11            {
12              communication cost(service v,service j)
+=communication cost(service v ,service j in DAG i);
13                Meet (service v,service j) +=1;
14                    }
15          RankCooparator(service v,service
j)=Norm(communicationcost (service v,service j))+Norm(Meet
(service v,service j));
16          insert service j to list;
17       }
18   sort the list with RankCooparator
19   return (first item from list)
20 {
```

۳-۳ مرحله ســـوم: انتخاب مقصد مناســـب برای مهاجرت خدمت

در این مرحله باید میزبانی مناســـب برای نگاشـــت خدمت منتخب یافت شـــود. در روش پیشنهادی ابتدا از لیســـت میزبان‌ها، میزبان‌هایی که بیش از حد آستانه[۳] [۱۸] انـــرژی مصـــرف می‌کنند، حذف می‌شـــوند. حال باید از لیســـت میزبان‌های نامزد، میزبانی انتخاب شود که شـــامل خدمت‌های همکار خدمت منتخب با استفاده از رابطه (۲) اســـت، چنانچه میزبان‌هایی یافت شـــدند کـــه به یک میزان حاوی خدمت همـــکار بودند، میزبانی انتخاب می‌شود که در آن مجموع اولویت خدمت‌هایش کم باشـــد. چنانچه با تمام این شـــرایط هنوز خدمت به میزبانی نگاشت نشده اســـت به اولین میزبان نگاشت می‌شود.

برای انتخاب میزبان مناسب باید ابتد از لیست میزبان‌ها، میزبان‌هایی که در زمـــان MinEST(v) و MaxEFT(v)، در دســـترس هســـتند در لیســـت نامزد میزبان‌ها درج شوند. سپس برای تک تک میزبان‌های موجود در لیست بررسی می‌شـــود، چنانچه میزبان به حد آســـتانه رسیده باشد از لیست میزبان‌های نامزد حذف می‌شود در غیر این صورت بررسی می‌شود چنانچه آن میزبان بیشترین تعداد خدمت همکار را داشـــت انتخاب می‌شـــود و چنانچه میزبان‌هایی یافت شـــوند که تعداد خدمت‌های همکار در آن‌ها بیشترین هستند ولی مقدار آن در چند میزبان یکسان است میزبانی انتخاب می‌شـــود کـــه مجمـــوع اولویت خدمت‌هـــا در آن کمتر باشـــد. در نهایت اگر خدمت بـــه میزبانی اختصاص نیافت، لیســـت میزبان‌های نامزد بر اساس مجموع اولویت خدمت‌های هر میزبان، به صورت نزولی مرتب می‌شـــود و خدمت به اولین میزبان موجود در این لیست اختصاص می‌یابد.

Input: Service v, Service v’
Output: A mapping of Service v to Host y
Select Host (Service v, Service v’)

3- Threshold

جدول ۱: مشخصات میزبان شبیه‌سازی

HOST_TYPES	HP ProLiant ML110 G4	HP ProLiant ML110 G5
HOST_MIPS	1860	2660
HOST_PES	2	2
HOST_RAM	4096	4096
HOST_BW	(1 Gbit/s) 1000000	(1 Gbit/s) 1000000
HOST_STORAGE	1000000(1 GB)	1000000(1 GB)

```
    {
1           allocatedHost=null;   maxCountV'=0;   sumRank=0;
2         foreach host
3         {
4         if  each host available Time  between  MinEST(v) and
MaxEFT(v), insert to Candidate Host List
5         }
6          foreach Host y in Candidate Host List
7         {
8         if ( T_Host y > threshold)
9         {
10        remove Host y from Candidate Host List;
11        }
12        else
13        {
14        if  (Host y have v')
15        {
16        if  (maxcountv' < count v' in Host y)
17        {
18        allocatedHost= Host y
19        maxcountv'=count v' in Host y
20        }
21        else if(maxcountv'== count v' in Host y)
22        {
23        if (sumRank allocatedHost > SumRankHost y)
24        {
25        allocatedHost= Host y
26        }
27        }
28        }
29        if (allocatedHost !=null )
30        {
31        Migration v to allocatedHost;
32        SumRank of allocatedHost += Rank(v);
33        }
34        else
35        {
36                                          sort Candidate Host
List With SumRank of Host (order by Descending)
37                                          Migration v to first
Host y in Candidate Host List;
38                                          SumRank of Host
y + = Rank(v);
39        }
40
}//end if in line 14
41                                  }// end if in line 8
42        } // end Function
```

۴- ارزیابی روش پیشنهادی

٤-١ شبیه‌سازی و آزمایش

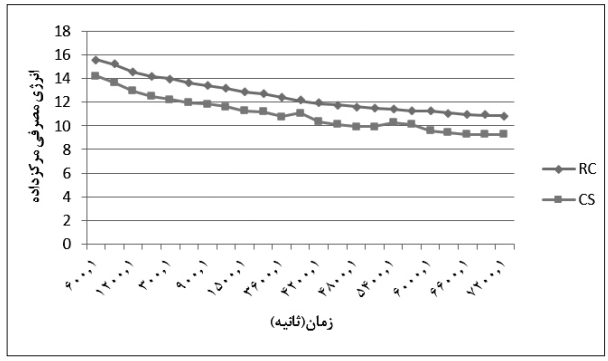
شبیه‌ســـازی و آزمایش توسط کد منبع کلودسیم[4] انجام شده است. کلودســـیم یک چارچوب قابل توسعه و متن‌باز را بـــا مدلی یکپارچـــه به منظور آزمایش‌ها و شبیه‌ســـازی زیرساخت‌های محیط ابری و خدمت‌ها ارائه نموده است [۱۹]. هدف اصلی تعیین میزان مصرف انرژی است. در مطالعات [۲۰] و [۲۱] مصـــرف برق توســـط خدمت دهنده می‌تواند با دقت توســـط یک رابطه خطی بین مصرف انرژی و استفاده از پردازنده توصیف شود. برای آزمایش‌ها از داده‌های ارائه شـــده در پلنت لب ۲۰۱۱۰۳۰۳ در [۱۸] اســـتفاده شده است. که یک مرکـــز داده حاوی ۸۰۰ گره میزبان فیزیکی که نیمی از آن HP ProLiant ML110 G4 و بقیـــه HP ProLiant ML11 درکـــزداه ان فیزیکی که نیمی از آن HP ProLiant ML110 G5 می‌باشـــند. خصوصیات آن‌ها در جدول (۱) نشان داده شده اســـت و شامل ۱۰۵۶ ماشـــین مجازی با ویژگی‌هایی که در جدول (۲) مشاهده می‌شود.

در ابتدای شبیه‌ســـازی لیســـتی از میزبان‌ها و لیســـتی از خدمت‌ها، ســـپس همکاری خدمت‌ها بـــا هم و گردش کار آن‌ها به صورت تصادفی ساخته می‌شود. تعداد کل گردش کارها به اندازه ۱۰ اســـت. سپس سیاستی که باید خدمت را انتخاب کند و هچنین سیاســـتی که خدمت منتخب را به یک میزبان مناسب نگاشت نماید، ساخته می‌شود. در طول زمان شبیه‌سازی الگوریتم تشخیص خدمت‌های همکار و مهاجرت آن‌ها در هر ۳۰۰ میلی ثانیه و یا زمانی که درخواست جدیدی از راه برســـد اجرا میشود. همچنین نتایج این شبیه‌سازی با

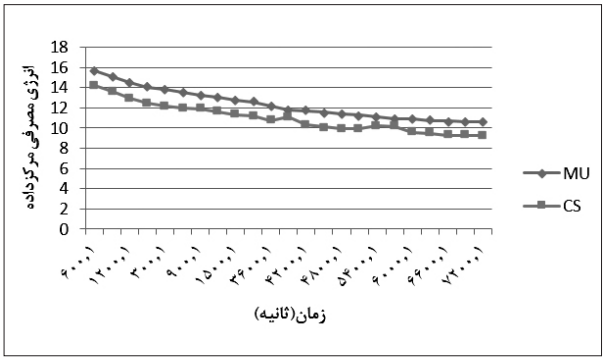
4- Cloudsim

جدول ۲: مشخصات ماشین‌های مجازی شبیه‌سازی

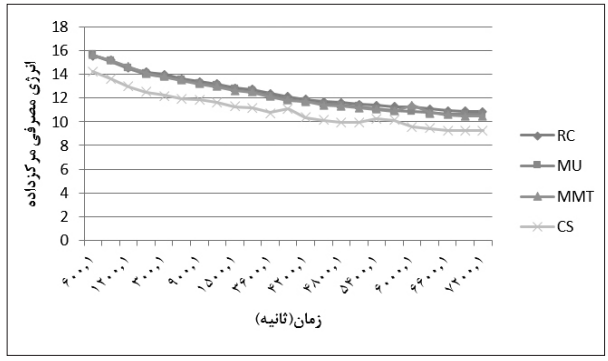
VM_TYPES	A	B	C	D
VM_MIPS	2500	2000	1000	500
VM_PES	1	1	1	1
VM_RAM	870	1740	1740	613
VM_BW	100000 (100 Mbit/s)	100000 (100 Mbit/s)	100000 (100 Mbit/s)	100000 (100 Mbit/s)
VM_SIZE	2500 (2.5 GB)	2500 (2.5 GB)	2500 (2.5 GB)	2500 (2.5 GB)



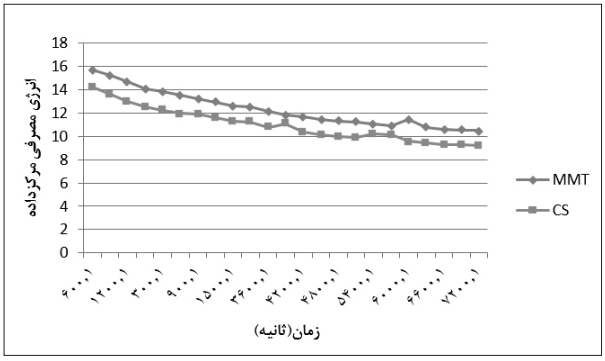
شکل ۱: انرژی مصرفی روش پیشنهادی و روش MMT



شکل ۲: انرژی مصرفی روش پیشنهادی و روش RC



شکل ۳: انرژی مصرفی روش پیشنهادی و روش MU



شکل ۴: انرژی مصرفی روش پیشنهادی با روش‌های دیگر

سیاست‌های موجود در [۱۸] مقایسه شده است.

۴-۲ نتایج شبیه‌سازی

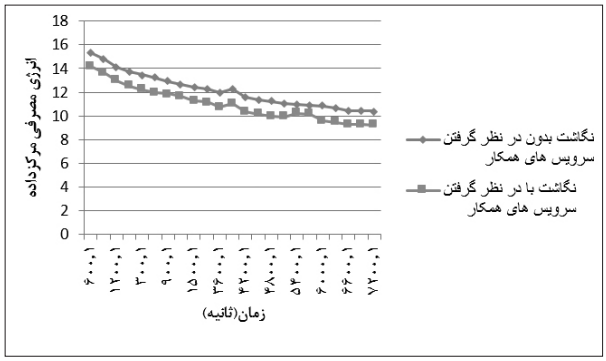
روش پیشنهادی خدمت همکار[5](CS) با سه الگوریتم زیر که بهترین الگوریتم‌های موجود و مناسب‌تر با موضوع پژوهش می‌باشند، مقایسه شده است. با توجه به این‌که در روش پیشنهادی از مهاجرت زنده استفاده می‌شود. الگوریتم‌های ارائه شده نیز راهکاری برای کاهش انرژی مراکز داده با استفاده از فرآیند مهاجرت زنده و تخصیص خدمت‌ها به میزبان می‌باشند.

- حداقل زمان مهاجرت[6] (MMT) [۲۲].
- سیاست انتخاب تصادفی[7](RC) [۱۸].
- حداقل مصرف (استفاده)[8] (MU) [۲۳].

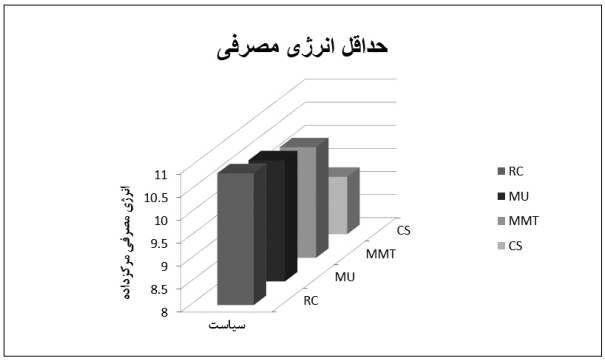
در سیاست MMT که در واقع حداقل زمان مهاجرت می‌باشد، برای انتخاب یک خدمت، خدمتی انتخاب می‌شود که زمان تکمیل مهاجرت و اختصاص به میزبان، در آن حداقل باشد. در نمودارها محور افقی، زمان (ثانیه) و محور عمودی، مصرف انرژی مرکز داده را نمایش می‌دهند. همان‌طور که در شکل (۱) مشاهده می‌شود روش پیشنهادی در مقایسه با روش MMT انرژی مصرفی کاهش پیدا کرده است.

در سیاست RC که در واقع سیاست انتخاب تصادفی می‌باشد، برای انتخاب یک خدمت، خدمتی به صورت تصادفی

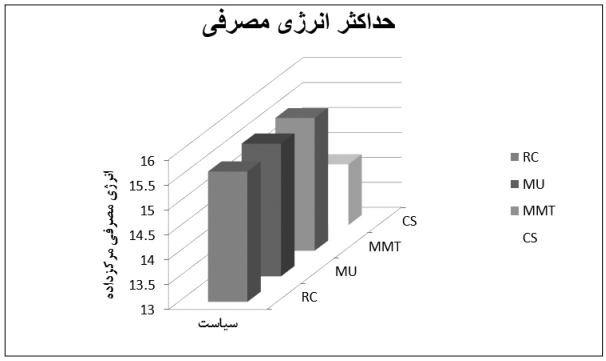
5- Cooperator Service
6- Minimum Migration Time
7- Random Choice Policy
8- Minimum Utilization



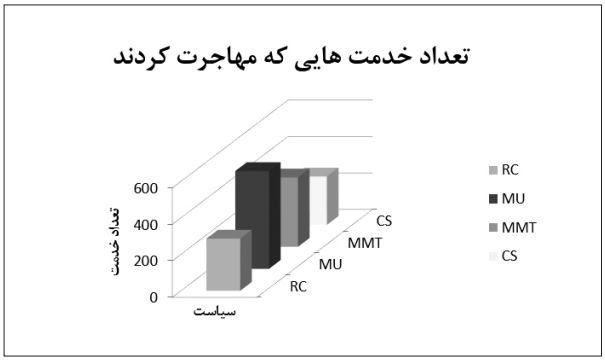
شکل ۵: حداقل انرژی مصرفی



شکل ۶: حداکثر انرژی مصرفی



شکل ۷: نگاشت با/بدون در نظر گرفتن خدمت‌های همکار در روش پیشنهادی



شکل ۸: تعداد خدمت‌هایی که مهاجرت نمودند

انتخاب می‌شود. همان‌طور که در شکل (۲) مشاهده می‌شود روش پیشنهادی در مقایسه با روش RC انرژی مصرفی کاهش پیدا کرده است.

در سیاست MU که در واقع حداقل مصرف (استفاده) می‌باشد، برای انتخاب یک خدمت، خدمتی انتخاب می‌شود که نیاز به حداقل استفاده از پردازنده نسبت به دیگر خدمت‌ها در میزبان دارد. همان‌طور که در شکل (۳) مشاهده می‌شود روش پیشنهادی در مقایسه با روش MU انرژی مصرفی کاهش پیدا نموده است.

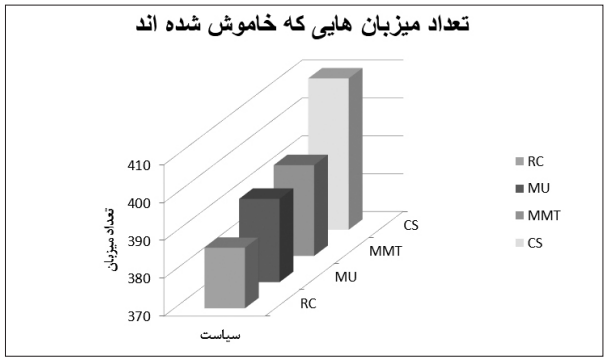
شکل (۴) هر چهار سیاست را در یک نمودار نشان می‌دهد روش پیشنهادی نسبت به سه سیاست دیگر، در مرکز داده انرژی کمتری مصرف شده است.

در هر چهار سیاست در بازه‌های زمانی که شبیه‌سازی صورت گرفته است حداقل انرژی مصرفی مرکز داده استخراج شده است، همان‌طور که در شکل (۵) دیده می‌شود، سیاست روش پیشنهادی کمترین حداقل انرژی مصرفی را دارد.

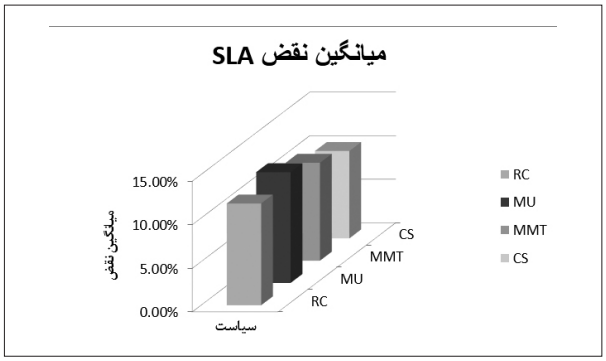
در هر چهار سیاست در بازه‌های زمانی که شبیه‌سازی صورت گرفته است حداکثر انرژی مصرفی مرکز داده استخراج شده است، همان‌طور که در شکل (۶) دیده می‌شود، سیاست روش پیشنهادی کمترین حداکثر انرژی مصرفی را دارد.

اکنون می‌خواهیم مقایسه را بدین گونه انجام دهیم که خدمت با روش پیشنهادی انتخاب شود ولی در مسئلهٔ تخصیص آن به میزبان، یکبار به میزبانی که حاوی خدمت‌های همکار است تخصیص یابد و یکبار به اولین میزبانی که به حد آستانه نرسیده و میزان انرژی مصرفی آن کم بوده (طبق سیاست‌های گذشته)، بدون در نظر گرفتن این‌که آیا حاوی خدمت‌های همکار است یا خیر، تخصیص یابد و این دو حالت با هم مقایسه شود.

همان‌طور که در شکل (۷) مشاهده می‌شود، در روش پیشنهادی یعنی تخصیص به میزبان حاوی خدمت‌های همکار میزان انرژی مصرفی مرکز داده کمتر از حالتی است که



شکل ۹: تعداد میزبان‌هایی که خاموش شده‌اند.



شکل ۱۰: میانگین نقض SLA

خدمت منتخب برای مهاجرت به میزبانی اختصاص داده شد که وجود خدمت‌های همکار در آن در نظر گرفته نشده است.

در شـــکل (۸) تعداد خدمت‌هایی که در طول شبیه‌سازی مهاجرت کردند در روش پیشنهادی کمتر است و این نشان می‌دهد که در سیاست روش پیشنهادی خدمت‌های کمتری نیـــاز به مهاجرت پیدا می‌کننـــد و از آنجایی که هر چه تعداد مهاجرت‌ها کمتر باشـــد مرکز داده، انرژی کمتری مصرف می‌کنـــد. بنابراین با توجه بـــه نتیجه به‌دســـت آمده روش پیشنهادی منجر به کاهش مصرف انرژی مرکز داده می‌شود.

در شکل (۹) روش پیشنهادی در طول اجرای شبیه‌سازی، با قـــرار گرفتن خدمت‌هـــای همکار در کنـــار یکدیگر تعداد میزبان‌های بیشتری خاموش می‌شوند و هر چه میزبان‌های بیشتری خاموش باشـــند انرژی مصرفی مرکز داده کاهش می‌یابـــد. بنابراین با توجـــه به نتیجه به‌دســـت آمده روش پیشنهادی منجر به کاهش مصرف انرژی مرکز داده می‌شود.

در شـــکل (۱۰) روش پیشـــنهادی در طـــول اجـــرای شبیه‌سازی، با قرار گرفتن خدمت‌های همکار در کنار یکدیگر، میانگین نقض SLA کمتر می‌باشد.

۵- نتیجه‌گیری و کارهای آینده

با توجه به افزایش روزافزون محبوبیت رایانش ابری، اگر انرژی‌ای که در منابع ارائه دهنده خدمات آن مصرف می‌شود کنترل نگردد، آنـــگاه در گام اول هزینه ارائه خدمت افزایش می‌یابد و در پـــی آن روی هزینه پرداختی خدمت گیرندگان تأثیر خواهد گذاشـــت. مورد مهم‌تر این‌که این مســـئله سهم زیادی در افزایش آلودگی محیط زیســـت خواهد داشت. لذا کشف راهکارهای بهره‌وری انرژی بسیار حیاتی است. از این رو در این مقاله با ارائه روشـــی که خدمت‌های همکار را در کنار هم قرار می‌دهد، باعث کاهش مصرف انرژی شد.

برای کارهای آینده می‌توان به موارد زیر اشاره نمود:

- پیش‌بینی شروع همکاری خدمات ها

در این تحقیق به موضـــوع مهاجرت خدمات‌های همکار پرداخته شـــد، اما به خدمات‌هایی که ممکن است در آینده با هم همکاری داشته باشند اشاره‌ای نشد. می‌توان با استفاده از تاریخچه ارتباطات خدمات و با به‌کارگیری یکی از مدل‌های پیش‌بینی آینده بر اساس گذشته، خدماتی را که ممکن است در آینده با هم همکار شـــوند شناســـایی کـــرد و به یکدیگر مهاجرت داد.

- پیش‌بینی زمان پایان همکاری خدمات

می‌توان زمان پایان همکاری خدمات را به عنوان یکی از پارامترهای انتخاب خدمات برای مهاجرت در نظر گرفت. به این صورت که اگر پیش‌بینی شد که چند خدمت زمان زیادی با هم همکاری نخواهند داشـــت، انتخاب آن‌ها برای مهاجرت اولویت کمتری نسبت به خدماتی که مدت زمان بیشتری با هم همکاری خواهند داشت، پیدا می‌کند. برای این منظور می‌توان از تاریخچه مهاجرت‌های موفق و ناموفق خدمات اســـتفاده نمود.

- مهاجرت خدمات به نزدیکی مکان فیزیکی درخواســـت کننده خدمات

در این پژوهش خدمات همکار شناســـایی و به پیش هم

انتقال داده شـــد. می‌توان مهاجـــرت خدمات‌های همکار که پیش‌بینی می‌شـــود مدت زیادی با هم همکاری داشته باشند را به مکانی که از نظر فیزیکی بیشتر درخواست‌ها از نزدیکی آن ارســـال می‌شود، بررسی نمود و مشاهده نمود که چنین اقدامی موجب کاهش بیشتر زمان پاسخ خدمات می‌شود.

منابع

[1] Xiong Fu,Chen Zhou,"Virtual Machine selection and placement for dynamic consolidation in Cloud Computing environment " in Frountiers of Computer Science, Volume 9,Issue 2, Pages 322-330

[2] V.Vinothina,Dr.R.Sridaran,Dr Padmavathi Ganapathi,"A survey on resource allocation strategies in cloud computing" in International Journal of Advanced Computer Science and Aplications, Vol.3, No.6,2012

[3] Esfandiarpoor S,Pahlavan A, Goudarzi M. Structure-aware online virtual machine consolidation for datacenter energy improvement in cloud computing. Computers & Electrical Engineering, Volume 42,February 2015,Pages 74-89

[4] Raja Wasim Ahmada, Abdullah Gania, Siti Hafizah Ab. Hamida, Muhammad Shiraza,,Abdullah Yousafzaia, Feng Xiab,"A survey on virtual machine migration and server consolidation frameworks for cloud data center, Journal of Network and ,Computer Applications,Volume 52,June 2015,Pages 11-25

[5] Jipeng and Gaoming Tang, "Virtual Machine Placement Strategy Research " Cyber Enabled Distributed Computing and Knowledge Discovery (CyberC), 2013 Internatinal Conference on Beijing. 10-12 Oct. 2013 ,Pages 294-297

[6] Jian Wan, Fei Pan and Congfeng Jiang,"Placement Strategy of Virtual Machines Based on Workload Characteristics" Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International. Shanghai 21-25 May 2012, Pages 2140-2145

[7] Andrew J. Younge, Gregor von Laszewski, Lizhe Wang, "Efficient resource management for Cloud computing environments" in Green Computing Conference, 2010 International, Chicago,15-18 Aug.2010, Pages 357-364

[8] Nagpure, M.B., Dahiwale, P., Marbate,P.," An efficient dynamic resource allocation strategy for VM environment in cloud" in Pervasive Computing (ICPC),2015 International Conference on Pune 8-10 Jan.2015 Pages 1-5

[9] Xinying Zheng, Yu Cai,"Dynamic Virtual Machine Placement for Cloud Computing Environments" in 43rd Internatinal Conference on Parallel Processing Workshops on 9-12 Sept.2014 Pages 121-128

[10] Satora Ohta," Virtual Machine Placement Algorithms to minimize Physical Machine Count" in Network Operations and Management Symposium (APNOMS) Conference,2013 15th Asia-Pacific ,25-27 Sept.2013,Pages 1-3

[11] Atefeh Khosravi,Saurabh Kumar Gage, Rajkumar Buyya, "Energy and Carbon-Efficent Placement of Virtual Machines in Distributed Cloud Data Centers" in 19th International Conference,Aachen,Germany ,August.2013 Pages 26-30

[12] A. Beloglazov, J Abawajy,and R Buyya," Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing",Elsevier Future Generation Computer Systems. 2011,Pages 758-761

[13] M. Bichler, T. Setzer, B. Speitkamp. " Capacity Planning for Virtualized Servers". Presented at Workshop on Information Technologies and Systems (WITS),Milwaukee, Wisconsin, USA, 2006.

[14] N. Bobroff, A. Kochut, K.A. Beaty, " Dynamic placement of virtual machines for managing sla violations", in: Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management, IM'07, 2007.

[15] B. Speitkamp, M. Bichler, " A mathematical programming approach for server consolidation problems in virtualized data centers", IEEE Transactions on Services Computing, 2010.

[16] W.Shi, and B. Hong," Towards Profitable Virtual Machine Placement in the Data Center",IEEE,,International Conference on Utility and Cloud Computing. 2011, Pages138.

[17] R. Buyya, A. Beloglazov, J. Abawajy, " Energy-efficient management of data center resources for cloud computing": a vision, architectural elements, and open challenges, in: Proceedings of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2010) , Las Vegas, USA, July. 2010, Pages 12-15.

[18] Beloglazov. A and Buyya. R,"Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers", Concurrency and Computation: Practice and Experience (CCPE), ISSN: 1532-0626, Wiley Press, New York, USA,DOI:10.1002/cpe.1867, 1397–1420, 2011.

[19] Calheiros. R N, Ranjan. R, Beloglazov. A, Rose CAFD, Buyya. R," CloudSim: a toolkit for modeling and simulation of Cloud computing environments and evaluation of resource provisioning algorithms", Software: Practice and Experience, 41(1):23–50, 2011.

[20] Fan. X, Weber. WD, Barroso. LA," Power provisioning for a warehouse-sized computer. Proceedings of the 34th Annual International Symposium on Computer Architecture (ISCA 2007)", ACM New York, NY, USA; 13–23, 2007.

[21] Kusic. D, Kephart. JO, Hanson. JE, Kandasamy. N, Jiang. G," Power and performance management of virtualized computing environments via lookahead control", Cluster Computing 2009; 12(1):1–15, 2009.

[22] Cleveland WS. Robust locally weighted regression and smoothing scatterplots. Journal of the American statistical association 1979; 74(368):829–836.

[23] Gandhi A, Harchol-Balter M, Das R, Lefurgy C. Optimal power allocation in server farms. Proceedings of the 11th International Joint Conference on Measurement and Modeling of Computer Systems, ACM New York, NY, USA, 2009; 157–168.