

ارائه روشی جهت بهبود توازن بار در محاسبات ابری با استفاده از الگوریتم گرم شب تاب

سیده فرشته سبحانی*^۱، سید جواد میرعابدینی^۲

۱- دانشجو، ارشد مهندسی کامپیوتر، کامپیوتر، دانشکده فنی مهندسی، دانشگاه آزاد واحد تهران

جنوب، ایران

fereshte.sobhany68@gmail.com

۲- استادیار، مهندسی کامپیوتر، کامپیوتر، دانشکده فنی مهندسی، دانشگاه آزاد واحد تهران

مرکزی، ایران

j_mirabedini@iauctb.ac.ir

چکیده

هدف اصلی رایانش ابری، بهبود استفاده از منابع توزیع شده، ترکیب آنها جهت رسیدن به گذردهی بالاتر و قادر بودن به حل مشکلات محاسباتی در مقیاس بزرگ است. هم اکنون توازن بار در سیستم های رایانش ابری یک چالش می باشد و نیازمند یک راه حل توزیعی است. با انتقال پویای حجم کار محلی از یک ماشین به ماشینی دیگر و یا ماشینی که کمتر مورد استفاده قرار می گیرد، می توان توازن ایجاد کرد. در اینجا معیارهایی مانند زمان پاسخ، کارایی، گذردهی، انرژی مصرفی و انتشار کربن مورد ارزیابی قرار گرفته است. هدف اصلی الگوریتم توازن بار در محیط رایانش ابری بهبود زمان پاسخ کار بوسیله توزیع بار کلی سیستم است. الگوریتم های توازن بار که قالب اصلی آن توازن بار و بهبود استفاده از منابع محسوب می شود، می توانند با پارامترهای مختلفی که از مهم ترین آن ها ویژگی های مورد استفاده توسط مطلوبیت و میزان بهره وری می باشد، مورد بررسی قرار گیرند. در این پایان نامه، با بررسی پارامترهای توازن بار و با استفاده از الگوریتم گرم شب تاب به بهینه سازی زمان اجرا در یک مجموعه از وظایف رسیدیم. نتایج یافته های تحقیق نشان می دهد که روش پیشنهادی نسبت به روش ژنتیک توازن بهتری را برای زمان پاسخ و حافظه ارائه کرده است. یافته های تحقیق نشان دهنده رشد ۵ درصدی توازن بار در بهره وری پردازنده نسبت به الگوریتم ژنتیک می باشد.

کلمات کلیدی: رایانش ابری، توازن بار، الگوریتم گرم شب تاب

۱-مقدمه

محاسبات ابری یکی از جدیدترین تحولات در فناوری اطلاعات است و با مرور زمان فراگیر می شود. بستر محاسبات ابری یک بستر سرویس دهنده کاملاً اتوماتیک است که به کاربران اجازه خرید، ایجاد از راه دور، مقیاس پذیری پویا و مدیریت سیستم را می دهد. محاسبات ابری مدلی است برای فراهم کردن دسترسی آسان از طریق شبکه بر اساس تقاضای کاربر به مجموعه ای از منابع رایانشی قابل تغییر و پیکربندی مانند شبکه ها، سرورها، فضای ذخیره سازی، برنامه های کاربردی و سرویس ها که این دسترسی بتواند با کمترین نیاز به مدیریت منابع یا دخالت مستقیم فراهم کننده سرویس به سرعت فراهم شده یا آزاد گردد. محاسبات ابری دارای یک معماری سه لایه است که در لایه زیر ساخت با مدیریت منابع مواجه هستیم، در لایه بستر امکاناتی برای توسعه برنامه های کاربردی بر روی منابع ابری فراهم شده است و در لایه نرم افزار برنامه کاربردی برای استفاده توسط کاربر نهایی قرار می گیرد. چنانچه بخواهیم نحوه مدیریت منابع و زمانبندی انجام کارها را با توجه به مساله مورد نظر بهبود دهیم باید وارد لایه زیر ساخت شویم. اگر بخواهیم روی توسعه یک مسئله خاصی به صورت عمومی کار کنیم وارد لایه بستر می شویم و اگر بخواهیم روی نحوه ارائه سرویس نهایی به کاربر تمرکز کنیم باید وارد لایه نرم افزار شویم. هدف اصلی رایانش ابری بهتر کردن استفاده از منابع توزیع شده، ترکیب آن ها برای رسیدن به گذردهی بالاتر و قادر بودن به حل مشکلات محاسباتی در مقیاس بزرگ است [۲]. در حال حاضر توازن بار و قابلیت اطمینان در سیستم های

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

رایانش ابری یک چالش می باشند. همیشه یک راه حل توزیعی مورد نیاز می باشد، زیرا همواره نگهداری یک یا چند سرور بیکار و غیرفعال تنها برای انجام برخی از خواسته های مورد نیاز، امکان پذیر نمی باشد و یا اینکه به صرفه نیست. واضح است به دلیل مقیاس و پیچیدگی این سیستم ها انتساب متمرکز کارها به سرورهای خاص غیر ممکن است. برای مدیریت مناسب منابع فراهم کننده سرویس، به توازن بار نیازمندیم که به ارائه دهنده سرویس پیشنهاد می شود. توازن بار اصطلاحی عمومی است که برای توزیع بار پردازشی بزرگتر به گره های پردازشی کوچکتر برای بهبود کارایی کلی سیستم استفاده شده است. در یک محیط توزیع شده، توازن بار یک فرایند توزیع بار بین گره های مختلف سیستم های توزیع شده جهت بهبود سودمندی منابع و زمان پاسخ کار می باشد. ایده یک الگوریتم توازن بار باید از سر ریز شدن و بار کم هر گره خاص جلوگیری کند. در مورد یک محیط رایانش ابری انتخاب الگوریتم مربوطه آسان نیست، زیرا موانع اضافی مانند امنیت، قابلیت اعتماد، گذردگی و غیره را نیز شامل می شود. بنابراین هدف اصلی الگوریتم توازن بار در محیط رایانش ابری بهبود زمان پاسخ کار بوسیله ی توزیع بار کلی سیستم است. الگوریتم باید مطمئن شود که هیچ الگوریتم خاصی سرریز از بار نشود [۳] در این مورد الگوریتم های بسیاری وجود دارد. که ما برای انجام این تحقیق از یکی از الگوریتم های هوش گره جمعی به نام الگوریتم گرم شب تاب برای بهینه سازی مسئله مورد نظر بهره گرفته ایم.

۲- مروری بر کارهای مرتبط

در مرجع شماره [۱] در پژوهش حاضر با استفاده از الگوریتم کلونی مورچه جه بهبود تعادل بار در زیر ساخت و ایجاد نوآوری های جدید باعث بهبود بیشتر این الگوریتم شویم. نوآوری های نام برده شده مانند مهاجرت وظایف و پیش بینی اجرای وظایف مدنظر می باشد تا در کنار تعادل بار زمان پاسخ گویی و مصرف انرژی کاهش پیدا کند راهکار پیشنهادی با تعداد ۵۰ و ۱۰۰ مورچه صورت می گیرد. نتایج حاصل نشان می دهد زمان اجرا الگوریتم کاهش ۳۵ درصدی و مصرف انرژی بین ۱۰ تا ۱۵ درصد کاهش پیدا کرده است، در نتیجه راهکار پیشنهادی در زمان کمتری توانسته درخواست های موردنظر را اجرا کند، در واقع با کمک مهاجرت های درستی که روش پیشنهادی ارائه می دهد ماشین های که بار زیادی دارند با ماشین دیگری که بار کمتری دارند منتقل شوند و ماشین های که بار کمتری دارند نیز شناسایی شود و با خاموش کردن آن ها مصرف انرژی را کمتر کنیم.

در مرجع شماره [۲] یک الگوریتم نمونه برداری تصادفی بایاس شده، بر مبنای ایجاد گراف مجازی که دارای ارتباط بین همه ی گره های سیستم است، مطرح شده است، که در آن هر گره در گراف مرتبط با یک گره کامپیوتر در سیستم ابر است. این روش برای توازن بار سیستم ابر، دارای ویژگی مقیاس پذیری است. این روش همچنین قابل اطمینان بوده و روشی موثر برای توازن بار است، که عموماً برای توازن بار سیستم توزیع شده طراحی شده است.

در مرجع شماره [۳] تصمیم گیری می شود که کدامیک از انواع استراتژی های تحمل پذیری خطا به هر کار اعمال شود تا محاسبات قابل اعتماد و محدودیت کمتر شود. همچنین الگوریتم زمانبندی پیشنهاد شده (SAFT)، علاوه بر محدودیت کمتر، دارای کارآمدی ممتازی در بهبود نرخ خرابی کار نسبت به استراتژی انتخاب تحمل پذیری خطای ثابت دارد. زیرا استفاده از استراتژی تحمل پذیری خطای ثابت، برای زمانبندی کار بیش از حد از منابع استفاده می کند.

در مرجع شماره [۴] توسط bohar در سال ۲۰۱۵ یک الگوریتم از زمانبندی کار و تنظیمات پویا از بار گره ها درون سیستم گرید پیشنهاد شده است. درون یک سیستم محاسباتی توزیع شده در خواست های پردازش از کاربران به صورت تصادفی می رسند. یک برنامه ریزی خوب از این درخواست ها انتساب آن ها به پردازشگرهای موجود می باشد. بنابراین تمامی درخواست ها به سرعت ممکن پاسخ داده می شوند. به دلیل مشترک بودن منابع در سیستم های گرید یک زمانبندی پویا برای تعادل بار پیشنهاد می شود.

در مرجع شماره [۵] ایده این مقاله این است که بار کاری روی یک ماشین فیزیکی را به ماشین های مجازی بیکار موجود روی سرورهای فیزیکی دیگر انتقال می دهد و با خاموش کردن آن سرورها در مصرف انرژی برق صرفه جویی می کند. ایراد

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

این ایده این است که پیدا کردن یک ماشین فیزیکی دیگر و انتقال ماشین مجازی روی آن هم وقت گیر است و هم باعث صرف هزینه می شود. اینکه کدام ماشین مجازی برای انتقال مناسب تر است و آیا آن سرور امکانات سخت افزاری که این ماشین مجازی برای اجرا نیاز داشت را دارد یا نه، خود مسئله دیگری است. روشن کردن سروری که خاموش شده و به آن نیاز داریم خود شامل صرف زمان و هزینه می باشد.

در مرجع شماره [۶] یک الگوریتم خوشه بندی توضیح داده شده است که یک مفهوم از خوشه بندی در رایانش ابری را ارائه می دهد. در رایانش ابری الگوریتم های توازن بار زیادی موجود است. هر الگوریتم مزایا و معایب خودش را دارد. بسته به نیاز، یکی از الگوریتم استفاده می شود. کارایی الگوریتم می تواند با ایجاد یک خوشه بندی از گره ها افزایش پیدا کند. هر خوشه می تواند به عنوان یک گروه در نظر گرفته شود. فرآیند ایجاد خوشه بندی حول مفهوم جور کردن گره ها است. در این فرآیند ابتدا گره، یک گره همسایه را که گره متفاوتی است و جور کننده^۱ نامیده می شود را انتخاب می کند. گره جور کننده یک ارتباط با یک گره دیگر از نوع گره شروع کننده برقرار می کند. در نهایت گره جور کننده جدا می شود. این فرآیند تکرار می شود. کارایی سیستم با بالا بودن دسترس پذیری به منابع افزایش پیدا می یابد. این افزایش گذردهی به دلیل بهره برداری مؤثر از منابع است. کارایی سیستم با منابع بالا بهبود می یابد، بنابراین گذردهی نیز با استفاده مؤثر از این منابع، افزایش می یابد. همچنین با افزایش تنوع در سیستم کاهش می یابد.

در مرجع شماره [۷] یک راه حل بر اساس الگوریتم ژنتیک برای حل مشکل زمان بندی کارهای وابسته، فراهم آمده است. در این کار، از یک رویکرد بر پایه ارتفاع با روش تصادفی جهت ایجاد راه حل های اولیه، استفاده شده است. به منظور حفظ وابستگی بین کارها توابع گذار^۲ و جهش^۳ تعریف شده است. همچنین انتقال داده ها بین کارهای وابسته در نظر گرفته شده است. به منظور رعایت وابستگی در حین انجام جهش، از ارتفاع کارها استفاده شده است. این الگوریتم تعمیم یافته است و برای تمام کارهای وابسته برنامه های علمی قابل استفاده است.

در مرجع شماره [۸] یک الگوریتم که وابستگی نرم افزار در جریان کاری^۴ گراف جهت دار غیر مدور (DAG)^۵ را در بردار ارائه کرده است. هم زمان با قرار دادن یک ماشین مجازی برای کارها، یک تصمیم گرفته می شود که آیا همان ماشین مجازی برای بیشتر از یک کار در نظر گرفته شود یا خیر؟ این تصمیم بر اساس مبادله زمان و هزینه می باشد. در سطح یک کار، اگر زمان اجرا خیلی بالا باشد، بر کاهش آن تمرکز می شود. کاهش زمان اجرا می تواند با تعریف بیش از یک ماشین مجازی برای کارهایی که همان نوع ماشین مجازی را درخواست کرده اند، انجام شود که این ممکن است هزینه را بالا ببرد. از طرف دیگر اگر هزینه بالا رود، آن را با زمان بندی بیش از یک کار روی همان ماشین مجازی، کاهش می دهد که این زمان اجرای کل را افزایش می دهد. در این روش بین زمان و هزینه برای اینکه مقدار هیچ کدام خیلی بالا نرود و هر دو را در سطح حداقل نگه دارد، میان گیری^۶ می کند.

در مرجع شماره [۹] روش پیشنهادی Dam و سایرین از یک جدول شاخص شامل شناسه ماشین مجازی و درخواست های مرتبط با آن استفاده می کند. با ورود درخواست های جدید، زمان بندی را بر مبنای FCFs^۷ انجام و جدول را به روزرسانی می کند. اگر ماشین مجازی برای تخصیص درخواست موجود نباشد، تعداد تصادفی مورچه را با فرومون اولیه یکسان ایجاد می کند که به طور تصادفی شروع به حرکت می کنند. سپس بررسی می کند که آیا مورچه سفر خود را به اتمام رسانده یا نه و مقدار فرومون را به روزرسانی کرده و بررسی می کند که آیا راه حل بهینه است یا نه. در نهایت اگر همه مورچه ها تور خود را

¹ Matchmaker

² Crossover

³ Mutation

⁴ Workflow

⁵ Directed Acyclic Graph

⁶ Compromise

⁷ First Come First Served

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

کامل کرده باشند، همه بروز رسانی های فرمومون را مقایسه می کند تا بهترین راه حل خروجی را به دست آورد. این روش برای همه کارها اولویت یکسان در نظر گرفته است که سناریو حالت واقعی نمی تواند باشد و همچنین تحمل پذیری خطا را در نظر نگرفته است.

۳- روش معرفی شده

الگوریتم کرم شب تاب الهام گرفته شده از مطالعات و مشاهدات روی کرم شب تاب است. این مطالعات نشان داده که کرم های شب تاب حشرات اجتماعی هستند که در کلونی ها زندگی می کنند و رفتار آن ها بیشتر در جهت بقا کلونی هست تا در جهت بقا یک جزء از آن. یکی از مهم ترین و جالب ترین رفتار کرم های شب تاب ، رفتار آن ها برای یافتن غذا است و به ویژه چگونگی پیدا کردن کوتاهترین مسیر میان منابع غذایی و آشیانه.

بر اساس قواعد کرم شب تاب دو بخش اصلی الگوریتم کرم شب تاب وجود دارد، جذب کردن کرم شب تاب و حرکت به سوی کرم شب تاب جذب شده. جذابیت یک کرم شب تاب از شدت نور آن مشخص می شود. هر کرم شب تاب دارای جذابیت متمایز β می باشد که بر قدرت آن در جذب اجزای گروه دلالت دارد شکل تابع جذابیت یک کرم شب تاب ، تابع کاهشی و یکنواخت زیر می باشد

$$\beta(r) = \beta_0 e^{-\gamma r m}, (m \geq 1) \quad (1)$$

در این جا r فاصله ی میان دو کرم شب تاب است ، β_0 جذابیت در $r=0$ و γ یک ضریب جذب نور ثابت می باشد. حرکت کرم شب تاب i ، که توسط نمونه ی جذاب تر (روشن تر) خود یعنی کرم شب تاب j جذب شده است.

برای مسأله توازن بار، کاهش شدت روشنایی کرم شب بسیار اهمیت پیدا می کند به گونه ای که کاهش شدت روشنایی با فاصله از منبع آن متناسب است. همچنین نور در رسانه ها جذب می شود، بنابراین باید در نظر گرفت که جذابیت با میزان جذب تغییر می کند. فرمول زیر میزان شدت نور منبع مربوط به بهینه ترین کرم شب تاب را نشان می دهد.

$$I = I_s / r^2 \quad (2)$$

که I_s شدت نور منبع است. برای یک محیط با یک ضریب جذب نور ثابت γ ، شدت نور I با فاصله r تغییر می کند:

$$I = I_0 e^{-\gamma r} \quad (3)$$

که I_0 شدت نور اصلی می باشد.

سرور برای دریافت یک درخواست، جستجوی گره ها را برای یافتن گره های آزاد جستجو می کند. هنگامی که گره بدست می آید، آن را در لیست برنامه قرار می دهد و هنگامی که عنصر بالای صف پردازش می شود، گره آزاد به عنوان پاسخ به درخواست کاربر اختصاص داده می شود.

فلوچارت روش پیشنهادی نشان می دهد که در کدام بخش ها تغییرات مورد نیاز اعمال شده است. با توجه به فلوچارت مراحل مورد نظر برای الگوریتم پیشنهادی به شرح زیر هستند:

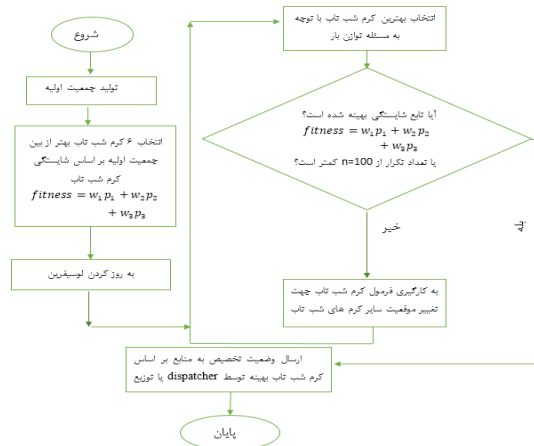
مرحله اول: دریافت اطلاعات ورودی: در این مرحله اطلاعات و داده هایی که قرار است مسئله بر روی آن اعمال شود تهیه و به سیستم داده می شود. این اطلاعات پهنای باند، طول صف، تعداد ماشین های مجازی و غیره هستند. چگونگی ورود این داده ها متفاوت است. در جاهایی که داده ورودی باشد که به صورت یک فایل اکسل به سیستم داده می شود و در جاهایی که پیکربندی نیاز باشد، پیکربندی و معماری مورد نیاز تهیه می شود.

مرحله دوم: تعیین ساختار کرم شب تاب: ساختار اولیه کرم شب تاب با دست کاری بر شدت بر روند اجرای الگوریتم آن تاثیر خواهد گذاشت. انتخاب پارامترهای مورد نیاز اجرا با تکرارهای متعدد صورت گرفته و به صورت آزمایش و خطا بوده است.

مرحله سوم: تعیین تابع fitness: در این مرحله بایستی یک تابع جهت بررسی درستی عملکرد و مقایسه بین حالت ها و در نهایت بهینه کردن تابع در نظر گرفت. این تابع هزینه ی اجرای الگوریتم کرم شب تاب را در هر دور نشان می دهد. در هر دور

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

پارامترهای بدست آمده از کرم شب تاب به این تابع داده می شود و بهترین جواب های ممکن بدست می آیند و در نهایت پارامترهای بدست آمده در دوره های بعدی مورد استفاده قرار می گیرند.



مرحله چهارم: تولید جمعیت اولیه: این مرحله به صورت تصادفی است به گونه ای که جمعیت اولیه و نحوه قرارگیری آنها در فضای مسئله به صورت تصادفی صورت می گیرد.

مرحله پنجم: انتخاب کرم شب تاب بهتر: در این مرحله ۶ کرم شب تاب بهتر جهت انجام عملیات انتخاب می شوند. انتخاب آنها بر اساس تابع ارزش می باشد.

مرحله ششم: اصلاح ساختار کرم شب تاب: در این مرحله با تغییر لوسیفیرین، تغییرات مکانی متناسب با دیگر کرم های شب تاب حاصل می شود. این مرحله به الگوریتم کرم شب تاب این قدرت را می دهد تا در دوره های متعدد بتواند بهتر و سریعتر به جواب بهینه برسد. توضیحات تغییرات لوسیفیرین در بخش مربوطه آورده شده است.

مرحله هفتم: انتخاب بر اساس مسئله توازن بار: در این مرحله کرم شب تاب هایی انتخاب می شوند که با توجه به توازن بار توانسته باشند تغییرات پارامتر بهتری را در نحوه حرکت و مکان خود ایجاد کنند. هرگاه یک کرم شب تاب پارامترهای خود مانند مکان را تغییر دهد، بایستی آن پارامترهای را جهت محاسبه توازن بار استفاده کرد. پارامترهایی که کرم تغییر می دهد، به فرمول محاسبه توازن بار داده می شود و محاسبه صورت می گیرد. هرچه توازن بهتری بدست آید درصد انتخاب کرم شب تاب بالاتر خواهد رفت. نحوه انتخاب آنها نیز به صورت تورنمنت می باشد به گونه ای که تمام پارامترها بر فرمول توازن بار اعمال کرده و سپس جواب های بدست آمده می شوند. آن جواب هایی که در تورنمنت بهتر باشند انتخاب خواهند شد.

مرحله هشتم: بررسی شرط پایان: شرط های پایان الگوریتم دو مورد هستند. مورد اول تعداد تکرار است که در این جا به صورت پیش فرض ۱۰۰ قرار داده شده است و موارد دیگر آن نیز مورد بررسی قرار گرفته است. شرط دوم بررسی اینکه آیا کرم های شب تاب توانسته اند مسئله توازن بار را بر روی ماشین های مجازی انتخابی حل کنند یا خیر. ممکن است در هر دور تغییراتی ایجاد شود، ولی تا زمانی که توازن بار بهینه بدست نیاید تعداد تکرار متوقف نخواهد شد.

مرحله نهم: تغییرات کرم شب تاب: در صورتی که شرایط پایان فراهم نباشد بایستی تغییرات مکانی کرم شب تاب مجدداً اعمال شود تا انتخابها صورت گیرد. در هر دور کرم های شب تاب با تغییر مکان خود، به سمت جواب بهینه حرکت می کنند.

۴- نتایج آزمایش و تحلیل نتایج

ارزیابی صورت گرفته براساس «تغییر در پارامترهای اولیه الگوریتم پیشنهادی» می باشد. در این بخش با اعمال تغییرات بر روی پارامترهای اولیه الگوریتم کرم شب تاب و ماشین های مجازی تغییرات بوجود آمده در اثر هر تغییر مشاهده و ثبت شده است.

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

جدول (۱): پارامترهای الگوریتم پیشنهادی

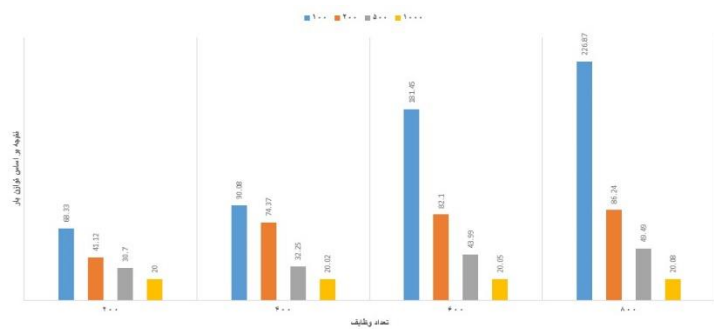
جمعیت اولیه	۲۵
γ	۱
β	۲
تکرارهای برنامه	متغیر

با توجه به پارامترها نتایج برنامه برای تعداد تکرارهای ۱۰۰، ۲۰۰، ۵۰۰ و ۱۰۰۰ به شرح زیر می باشد.

جدول (۲): نتایج برنامه برای تعداد تکرارهای مختلف

تعداد وظایف تعداد تکرار	۲۰۰	۴۰۰	۶۰۰	۸۰۰
۱۰۰	۳۳.۶۸	۰۸.۹۰	۴۵.۱۸۱	۸۷.۲۲۶
۲۰۰	۱۲.۴۱	۳۷.۷۴	۱۰.۸۲	۲۴.۸۶
۵۰۰	۷۰.۳۰	۲۵.۳۲	۹۹.۴۳	۴۹.۴۹
۱۰۰۰	۰۰.۲۰	۰۲.۲۰	۰۵.۲۰	۰۸.۲۰

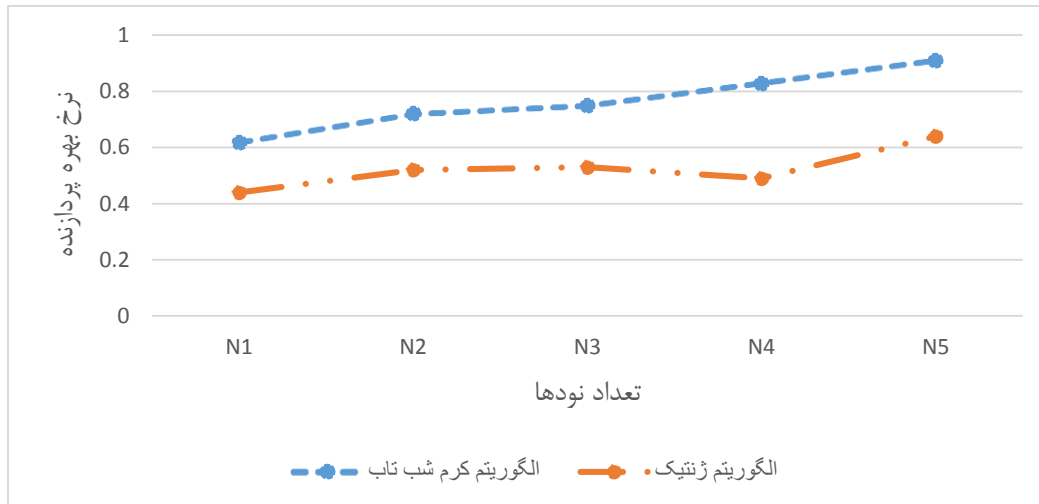
همانطور که در جدول بالا مشاهده می شود با تغییر در تکرار برنامه الگوریتم جوابهای بهتری را ارائه می دهد در نمودار ۱ مقایسه بین تکرارهای مختلف آورده شده است.



شکل (۱): نمودار تغییرات براساس تعداد تکرار

نمودار ۱ نشان دهنده تحلیل مقایسه ای از رویکرد پیشنهادی با رویکردهای معرفی شده می باشند. مقادیر رویکرد موجود از روش های زمان بندی متعادل بار حاصل شده از روش پیشنهادی نشان می دهد که رویکرد پیشنهادی با استفاده از نرخ پردازش بیشتر، موثرتر از رویکرد موجود در شرایط متعادل بار است. تجزیه و تحلیل نشان می دهد که استفاده از حافظه به طور متوسط از رویکرد پیشنهادی کمتر است. بنابراین، با توجه به شرایط متعادل بار و نرخ بهره CPU، رویکرد پیشنهادی دارای رفاه بیشتری نسبت به رویکرد موجود است.

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم



شکل (۱): نمودار مقایسه نرخ بهره پردازنده

۵- نتیجه گیری

نتایج شبیه سازی نشان داد که الگوریتم پیشنهادی به میزان بسیار زیادی به پارامترهای اولیه وابسته است که از میان پارامترهای اولیه الگوریتم، میزان تکرار الگوریتم به مراتب نتایج بهتری را به خروجی می برد. در واقع روش پیشنهادی با تعداد تکرار ۱۰۰۰، بهترین جواب که همان حداقل میزان make span است را به ما می دهد. کاهش make span در بیکار نمودن ماشین های مجازی و برقراری تعادل بارکاری بیشتر در محیط ابر مؤثر است. همچنین الگوریتم پیشنهادی با الگوریتم های دیگر در زمینه زمان بندی کارها مقایسه شد که نتایج نشان داد الگوریتم پیشنهادی زمان اجرا، زمان تکمیل آخرین کار و روند همگرایی بسیار بهتری نسبت به الگوریتم های دیگر دارد. این بدان معنی است که روش پیشنهادی از دو روش دیگر مؤثرتر و دارای کارایی بالاتری است.

۶- مراجع

- [1]. Y. Hu, B. Gong, and F. Wang, "Cloud model-based security-aware and fault-tolerant job scheduling for .computing grid," in ChinaGrid Conference (ChinaGrid), 2010 Fifth Annual, 2010, pp. 25–30
- [2]. Raval, A., Nasre, R., Kumar, V., Vadhiyar, S., & Pingali, K. (2017). Dynamic Load Balancing Strategies for Graph Applications on GPUs. ArXiv preprint arXiv: 1711.00231.
- [3]. Zhang, L., Chen, Y., Sun, R., Jing, S., & Yang, B. (2008). A task scheduling algorithm based on PSO for grid computing. International Journal of Computational Intelligence Research, 4(1), 37-43.
- [4]. Bhojar, A. A., & Dharmik, R. C. (2015). Design and Implementation of Job Scheduling in Grid Environment over IPv6. International Journal of Computer Science and Mobile Computing, 4(4), 243-250.
- [5]. Keerthika, P., & Suresh, P. (2015). A multiconstrained grid scheduling algorithm with load balancing and fault tolerance. The Scientific World Journal, 2015.
- [6]. Mahendiran, A., Saravanan, N., Subramanian, N. V., & Sairam, N. (2012). Implementation of K-means clustering in cloud computing environment. Research Journal of Applied Sciences, Engineering and Technology, 4(10), 1391-1394.
- [7]. Lagwal, M., & Bhardwaj, N. (2017, June). Load balancing in cloud computing using genetic algorithm. In Intelligent Computing and Control Systems (ICICCS), 2017 International Conference on (pp. 560-565). IEEE.
- [8]. Katyal, M., & Mishra, A. (2014). A comparative study of load balancing algorithms in cloud computing environment. ArXiv preprint arXiv: 1403.6918.
- [9]. Dam, S., Mandal, G., Dasgupta, K., & Dutta, P. (2014). An ant colony based load balancing strategy in cloud computing. In Advanced Computing, Networking and Informatics-Volume 2(pp. 403-413). Springer, Cham.
- [10]. Nitika, M., Shaveta, M., & Raj, M. G. (2012). Comparative analysis of load balancing algorithms in cloud computing. International Journal of Advanced Research in Computer Engineering & Technology, 1(3), 120-124.

- [11]. Krishna, P. V. (2013). Honey bee behavior inspired load balancing of tasks in cloud computing environments. *Applied Soft Computing*, 13(5), 2292-2303.
- [12]. Navimipour, N. J., & Milani, F. S. (2015). Task scheduling in the cloud computing based on the cuckoo search algorithm. *International Journal of Modeling and Optimization*, 5(1), 44.
- [13]. Dos Santos, M. J., & de M Fagotto, E. A. (2015). Cloud computing management using fuzzy logic. *IEEE Latin America Transactions*, 13(10), 3392-3397.
- [14]. Kaur, G., & Kaur, K. (2017). An Adaptive Firefly Algorithm for Load Balancing in Cloud Computing. In *Proceedings of Sixth International Conference on Soft Computing for Problem Solving* (pp. 63-72). Springer, Singapore.
- [15]. Yao, J., & He, J. H. (2012, April). Load balancing strategy of cloud computing based on artificial bee algorithm. In *Computing Technology and Information Management (ICCM), 2012 8th International Conference on* (Vol. 1, pp. 185-189). IEEE.
- [16]. Ramezani, F., Lu, J., & Hussain, F. K. (2014). Task-based system load balancing in cloud computing using particle swarm optimization. *International journal of parallel programming*, 42(5), 739-754.
- [17]. Maruthanayagam, D. D., & Prakasam, T. A. (2014). Job Scheduling in Cloud Computing using Ant Colony Optimization. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* Volume, 3.
- [18]. Tawfeek, M. A., El-Sisi, A., Keshk, A. E., & Torkey, F. A. (2013, November). Cloud task scheduling based on ant colony optimization. In *Computer Engineering & Systems (ICCES), 2013 8th International Conference on* (pp. 64-69). IEEE.
- [19]. Grover, J., & Katiyar, S. (2013, August). Agent based dynamic load balancing in Cloud Computing. In *2013 International Conference on Human Computer Interactions (ICHCI)* (pp. 1-6). IEEE.
- [20]. Mishra, S., & Patidar, K. AN IWRR METHOD BASED ON EFFICIENT LOAD BALANCING IN CLOUD COMPUTING.
- [21]. Hasan, R. A., & Mohammed, M. N. (2017). A Krill Herd Behaviour Inspired Load Balancing of Tasks in Cloud Computing. *Studies in Informatics and Control*, 26(4), 413-424.