دوفصلنامه محاسبات و سامانه های توزیع شده
سال هشتم، شماره۲، شماره پیاپی۱۶، صفحه ۱۱۴ تا ۱۳۰، سال ۱۴۰۴

ارائه یک مدل زمانبندی کارها در شبکه‌های گرید مبتنی بر الگوریتم بهینه سازی شیر

معصومه اکبری امامی
کارشناسی ارشد مهندسی نرم افزار گرایش شبکه‌های کامپیوتری دانشگاه آزاد اسلامی واحد زنجان

چکیده

چالش تعادل بار در رایانش ابری برای مدیریت موثر منابع ابر بسیار حائز مهم است و این مستلزم توزیع ترافیک شبکه ورودی یا بار کاری محاسباتی بین سرورهای متعدد است به شکلی که هیچ سروری بیش از حد بارگذاری نشود، از این رو بهبود استفاده از منابع، افزایش توان عملیاتی و کاهش زمان پاسخگویی تعادل بار برای سیستم‌های توزیع شده برای دستیابی به دسترسی بالا و تحمل خطا حیاتی می‌باشد. در این پژوهش یک مدل زمانبندی کارها در شبکه‌های گرید مبتنی بر الگوریتم بهینه سازی شیرارائه میشود. هدف این تحقیق ارائه یک رویکرد فراابتکاری بهینه‌سازی شیر بهبود یافته الگوریتم شیر است که برای حل مسائل تعادل بار در ابر ایجاد می‌شود. در واقع روش پیشنهادی چالش مشکل زمان‌بندی تک ماشینی را بررسی میکند که در آن ماشین باید پس از یک بازه دوره‌ای ثابت نگهداری شود. هدف این است که کل انحراف مطلق زمان‌های تکمیل برای مشکل بررسی شود به حداقل برسد. از این رو برای حل این مشکل از یک الگوریتم فراابتکاری جدید، یعنی الگوریتم بهینه‌سازی شیر، در روش پیشنهادی استفاده شد که نتایج مقایسه برتری مطلق الگوریتم پیشنهادی ما را به نسبت مقاله پایه را نشان داد.

واژه‌های کلیدی: زمانبندی کارها، شبکه‌های گرید و الگوریتم بهینه سازی شیر.

تاریخچه مقاله:
تاریخ ارسال: ۱۴۰۴/۰۷/۱۵
تاریخ اصلاحات: ۱۴۰۴/۰۸/۲۰
تاریخ پذیرش: ۱۴۰۴/۱۱/۲۰
تاریخ انتشار: ۱۴۰۴/۱۱/۲۹
ایمیل نویسنده مسئول:m.akbariiraninsurance@gmail.com

۱ - مقدمه و بیان مسئله

در محاسبات گرید، تعادل بار نقشی حیاتی در تخصیص مجدد مشاغل کاربر در زمانی که منابع شبکه بیش از حد بارگذاری می‌شوند، ایفا می‌کند. در چند سال گذشته، نمرات استراتژی های متعادل کننده بار توسط محققان برای بهبود زمان پاسخ، سربار ارتباط، توان عملیاتی و استفاده از منابع پیشنهاد شده است. در واقع مدیریت منابع یک موضوع چالش برانگیز در محیط شبکه است زیرا شامل مدیریت منابع شبکه توزیع شده در سراسر جهان است. منابع به گونه‌ای تخصیص داده می‌شوند که به طور جمعی مشکلی را که دارای الزامات عملکرد بالایی در طول پردازش است، حل کنند. با توجه به ماهیت پویای منابع شبکه، مدیریت منابع مشکل پیچیده تخصیص مجدد منابع را حل می‌کند، بنابراین کار کاربر به دریافت زمان پاسخ بهتر در یک محیط شبکه ادامه می‌دهد. مکانیسم متعادل کننده بار راه حل مناسبی است که منابع شبکه را به طور موثر با توجه به تغییرات استفاده از منابع شبکه تخصیص مجدد می‌دهد[۱]. گرید به عنوان یک پارادایم جدید برای حل مشکلات در علم، مهندسی، صنعت و تجارت در حال ظهور است. تعداد فزاینده‌ای از برنامه‌ها از زیرساخت گرید برای برآوردن نیازهای محاسباتی، ذخیره‌سازی و سایر نیازهای خود استفاده می‌کنند. یک سایت واحد دیگر نمی‌تواند تمام نیازهای منابع برنامه‌های کاربردی امروزی را برآورده کند و استفاده از منابع توزیع شده می‌تواند مزایای زیادی را برای کاربران برنامه به همراه داشته باشد. استقرار سیستم‌های گرید شامل مدیریت کارآمد منابع ناهمگن، توزیع شده جغرافیایی و پویا در دسترس است.با این حال، اثربخشی یک محیط گرید تا حد زیادی به اثربخشی و کارایی زمان‌بندی‌های آن بستگی دارد که به عنوان واسطه‌های منابع محلی عمل می‌کنند. زمان‌بندی شبکه به عنوان فرآیند نگاشت مشاغل گرید به منابع در چندین حوزه اداری تعریف می‌شود. یک کار گرید را می‌توان به بسیاری از وظایف کوچک تقسیم کرد. زمانبندی مسئولیت انتخاب منابع و زمانبندی کارها را دارد به گونه ای که نیازهای کاربر و برنامه از نظر زمان اجرای کلی (خروجی) و هزینه منابع مورد استفاده برآورده شود. سیستم‌های گرید معمولاً توسط مردم عادی استفاده نمی‌شوند، زیرا در نتیجه عدم آگاهی کافی است و همچنین باید به ارائه‌دهندگان منابع انگیزه‌های بیشتری برای تشویق به اشتراک گذاری منابع بیشتر و قابل اعتماد داده شود. تخصیص مشترک منابع برای برنامه های کاربردی چند سایتی

یا گردش کار مورد توجه بسیاری از افراد است، یعنی ارتباط بین دو یا چند سیستم شبکه ای (تعامل بین شبکه ای) به طور موثر، اما این تنها زمانی امکان پذیر است که ارتباطات درون شبکه ای کامل تر شود. همچنین برای جلوگیری از اتلاف زمان و پول و امکان مدیریت کارآمد انتقال منابع از یک کار به کار دیگر، نیاز به برنامه‌ریزی مناسب وظایف وجود دارد، همچنین به یک کار اجازه می‌دهد از چندین منبع به طور همزمان استفاده کند و تمام فعالیت‌های شبکه را بدون انجام کار پیگیری کند. سیستم کند و ناکارآمد انصاف در رسیدگی به همه کارها به گونه‌ای که همه کارها در یک زمان قابل اجرا باشد[۲].

زمان‌بندی کار پویا در محاسبات شبکه‌ای یک مسئله NP-hardاست که می‌توان با استفاده از تکنیک‌های مختلف رده زیر بهینه تعاونی آن را حل کرد. الگوریتم های تکاملی یکی از راه حل‌های زیر بهینه تعاونی است که می‌تواند نیازهای در حال تغییر پویا یک محیط ناهمگن مانند شبکه های محاسباتی را برآورده کند: مسائل باز عبارتند از: تولید محاسبات با توان بالا و ارزیابی معیارهای مختلف هزینه مانند زمان ساخت، کارایی شبکه و نسبت خطای کار. از این رو در واقع این پژوهش به این سوال پاسخ داده می‌شود که: " ارائه یک مدل زمانبندی کارها در شبکه‌های گرید مبتنی بر الگوریتم بهینه سازی شیر" امکان‌پذیر است؟

۲ - کارهای مرتبط

در مقاله‌ای تخصیص و زمان‌بندی منابع در محاسبات شبکه- استراتژی‌های جدید برای تخصیص و زمان‌بندی ارائه شد. آنها معتقدند که محاسبات گرید به یک زمینه مهم تحقیقاتی تبدیل شده است که از محاسبات توزیع شده معمولی و محاسبات با عملکرد بالا برای حل مسائل در مقیاس بزرگ تکامل یافته است. کاربردهای علمی و تجاری بسیار پیچیده هستند و به قدرت محاسباتی و فضای ذخیره سازی عظیم نیاز دارند. محیط محاسبات شبکه از فناوری برای اجرای برنامه های کاربردی در مقیاس بزرگ در منابع پشتیبانی می کند .تخصیص منابع و زمان‌بندی وظایف به دلیل برخی ویژگی‌های مطلوب مانند استفاده بهینه از منابع، بهبود در به حداقل رساندن زمان انتظار، زمان تکمیل کل و زمان پاسخگویی کل و هزینه کل منابع، توجه زیادی را از سوی جامعه تحقیقاتی در محاسبات گرید به خود جلب کرده است. زمان‌بندی وظایف برای منابع محاسباتی ناهمگن توسط بسیاری از دانشمندان مورد بررسی قرار گرفته و به عنوان یک مسئله کامل چندجمله‌ای غیر قطعی شناخته شده‌است[۳].

در مقاله‌ای دیگر برنامه ریزی شبکه با در نظر گرفتن مدیریت مصرف انرژی و کیفیت خدمات ارائه شد. در این تحقیق، مشکل زمان‌بندی شبکه به منظور به حداکثر رساندن سود با توجه به تکنیک مقیاس‌بندی ولتاژ و فرکانس دینامیکی، کیفیت خدمات مشتری محور و قیمت‌گذاری انرژی وابسته به زمان بررسی شده است. برنامه‌ریزی خطی عدد صحیح مختلط، برنامه‌ریزی محدودیت، یک الگوریتم اکتشافی حریصانه به همراه یک روش ترکیبی از الگوریتم ژنتیک و برنامه‌نویسی محدودیت توسعه داده شده‌اند. برخی از تکنیک‌ها برای بهبود کارایی مدل برنامه‌ریزی محدودیت ارائه شده پیشنهاد شده‌اند و اثربخشی آنها با استفاده از یک آزمایش فاکتوریل کامل بررسی می‌شود. پارامترهای الگوریتم ترکیبی پیشنهادی با آزمون تاگوچی تنظیم شده است. الگوریتم فراابتکاری ترکیبی، با زمان اجرای کوتاه، راه حل هایی در حدود ۱۸ درصد و ۸۸ درصد بهتر از بهترین راه حل مدل برنامه نویسی محدودیت برای نمونه های مسئله در مقیاس بزرگ ایجاد می‌کند. نتایج نشان می‌دهد که در صورت در نظر گرفتن اشتباه قیمت برق با نرخ ثابت در طول فرآیند برنامه ریزی، سود نهایی حدود ۲۲ درصد کاهش می-یابد[۴].

برنامه‌ریزی آگاه از شکست مبتنی بر گردش کار در محاسبات شبکه دز مقاله ای ارائه شد که نویسندگان معتقدند که زمان بندی به طور کلی یک مشکل دشوار است زیرا یک مسئله-NP کامل است .این درست است چه در گرید یا در هر محیط دیگری انجام شود. هنگامی که وظایف به یکدیگر وابسته هستند، مشکل پیچیده تر می شود. مسئله NP-کامل یک اکتشافی از پیش تعیین شده برای توصیف آنها ندارد. این امکان وجود دارد که یک اکتشافی خاص در برخی شرایط به خوبی عمل کند اما در شرایط دیگر نه، و این امر زمان‌بندی را حیاتی‌تر و حیاتی‌تر می‌کند. با هدف بهبود عملکرد برنامه و بهینه‌سازی خروجی حاصل، یک رویکرد زمان‌بندی آگاه از شکست مبتنی بر گردش کار[1] در این تحقیق برای حل مشکل زمان‌بندی برای وظایف وابسته در سیستم‌های مقیاس بزرگ مانند محاسبات شبکه پیشنهاد شده است .گردش کار تکلیف وابسته با گراف

[1] workflow based failure aware scheduling approach

غیر چرخه‌ای جهت‌دار[1] نشان داده می شود. روش "رویکرد زمان‌بندی آگاه از شکست مبتنی بر گردش کار "بر اساس مکانیسم تحمل خطای پست بازرسی افزایشی و اطلاعات خرابی منابع است .تجزیه و تحلیل نتایج نشان می‌دهد که روش پیشنهادی "رویکرد زمان‌بندی آگاه از شکست مبتنی بر گردش کار" ضمن افزایش قابلیت اطمینان و عملکرد سیستم، طول عمر و تعداد خرابی‌های سیستم را کاهش می‌دهد[۵].

در مقاله‌ای زمانبندی کار در محاسبه گرید با استفاده از اولین فیت، بهترین تناسب، و بدترین تناسب را ارائه می‌دهد. آنها معتقدند که محاسبات شبکه ای را می‌توان محاسبات خوشه‌ای توزیع شده در مقیاس بزرگ و پردازش شبکه توزیع شده موازی در نظر گرفت. دو موضوع مهم در مدیریت کار کاربر، تخصیص منابع و زمان بندی منابع مورد نیاز است. هنگامی که مشاغل کاربر ارسال می‌شوند، توسط واسطه‌های منابع مدیریت می شوند که منابع مناسب را پیدا و تخصیص می دهند. پس از مرحله تخصیص منابع، کار بر روی منابع موجود با توجه به منابع مورد نیاز کاربر برنامه ریزی می‌شود. در اکثر سیستم‌های شبکه‌ای با زمان‌بندی سنتی، مشاغل ارسال می‌شوند و در صف‌های اتاق انتظار قرار می‌گیرند تا منتظر بمانند تا منابع مورد نیاز در دسترس قرار گیرند. هر سیستم شبکه‌ای می‌تواند از یک الگوریتم زمان بندی متفاوت برای اجرای کارها بر اساس پارامترهای دیگر مانند منابع، زمان تحویل و مدت زمان استفاده کند. هیچ تضمینی وجود ندارد که این الگوریتم‌های زمان بندی سنتی کار را انجام دهند. استراتژی " خدمات به ترتیب ورود- زمان‌بندی سرویس در سوراخ چپ و راست"[2] استفاده از منابع را در یک سیستم شبکه با استفاده از یک زمانبندی محلی بهبود می‌بخشد. در مقایسه با استراتژی‌های سنتی، این تحقیق دو هدف دارد. ابتدا الگوریتم‌های اولین تناسب، بهترین تناسب و بدترین تناسب را مقایسه کنید تا زمان‌های خالی پیدا شده و آنها در یک نمای مجازی قرار می‌گیرد. دوم، مقدار زمان بیکاری را کاهش داده شده و نتایج نشان داد که روش "خدمات به ترتیب ورود- زمان‌بندی سرویس از سوراخ چپ و راست" می- تواند مقدار زمان بیکاری روش‌های "خدمات به ترتیب ورود- زودترین موعد اول"[3] و خدمات به ترتیب ورود را کاهش دهد.

زمان اجرای کلی اولین تناسب با استراتژی " خدمات به ترتیب ورود- زمان‌بندی سرویس در سوراخ چپ و راست" بهتر از "خدمات به ترتیب ورود-زودترین موعد اول" است[۶].

یک مقاله به استراتژی های زمان بندی شغلی در محاسبات شبکه ای پرداخته و نویسندگان معتقدند که محاسبات شبکه ای را می‌توان به عنوان محاسبات خوشه ای توزیع شده در مقیاس بزرگ و پردازش شبکه موازی توزیع شده در نظر گرفت .کاربران می‌توانند قدرت محاسباتی عظیمی را از طریق فناوری شبکه به دست آورند، که بدست آوردن آن از یک کامپیوتر تکی چالش برانگیز است. زمان‌بندی شغلی در محاسبات شبکه‌ای یک مسئله حیاتی است که بر قابلیت کلی سیستم شبکه تأثیر می‌گذارد. در زمان‌بندی سنتی، مشاغل در صف قرار می‌گیرند و منتظر در دسترس بودن منابع هستند. در صورتی که منابع مورد نیاز در زمان مشخص شده به دست نیامد، رزرو رد می شود. تأثیری که ایجاد می شود کاهش استفاده از منابع است. الگوریتم زمانبندی و پارامترهای مورد استفاده برای انجام کار ممکن است متفاوت باشد، مانند زمان اجرا، زمان تحویل و تعداد منابع. هیچ تضمینی وجود ندارد که کار با استفاده از الگوریتم زمان بندی اجرا شود. بنابراین، بهبود استفاده از منابع در سیستم شبکه و اطمینان از انجام کارها ضروری است. این مقاله یک استراتژی زمان‌بندی رزرو برای کار رابط ارسال پیام[4] پیشنهاد می‌کند ("ورودی اول سرویس گیری اول- زمان‌بندی سرویس از سوراخ چپ و راست"). کارهای رابط ارسال پیام به طور همزمان با استفاده از بیش از یک منبع برای پیاده سازی اجرا می شوند .پس از تکمیل، کارهای رابط ارسال پیام کاربر بر روی گره های محاسباتی مجازی برنامه ریزی شده و به گره‌های محاسباتی واقعی نگاشت می‌شوند. نتایج تجربی نشان می‌دهد که افزایش استفاده از منابع به شدت تحت تأثیر انعطاف‌پذیری زمانی است[۷]

الگوریتم موازنه بار دینامیکی مبتنی بر تخمین برای توزیع بار کارآمد و متعادل‌سازی در محیط محاسباتی شبکه ناهمگن نیز در مقاله‌ای ارائه شد. آنها معتقدند که برای تحقق بخشیدن به پتانسیل واقعی محاسبات گرید، مدیریت منابع نقش مهمی ایفا می‌کند. با این وجود، به دلیل ماهیت پویایی و

[1] directed acyclic grap
[2] FCFS-LRH: First Come First Serve Left Right Hole Scheduling
[3] FCFS-EDF: First Come First Serve Earliest Deadline First
[4] Message Passing Interface (MPI)

ناهمگونی در محاسبات گرید، مدیریت منابع شبکه با قابلیت توزیع و متعادل‌سازی بار مؤثر و کارآمد همچنان یک چالش است. در این مطالعه، یک الگوریتم موازنه بار پویا برای توزیع و متعادل‌سازی بار کارآمد در محیط محاسباتی شبکه ناهمگن پیشنهاد شده‌است. آزمایش‌های شبیه‌سازی گسترده برای ارزیابی اثربخشی الگوریتم پیشنهادی با استفاده از محبوب‌ترین شبیه‌ساز یعنی گریدسیم انجام می‌شود. نتایج مقایسه‌ای آزمایش‌های شبیه‌سازی نشان می‌دهد که رویکرد متعادل‌سازی بار پیشنهادی عملکرد برتری می‌دهد و از رویکردهای متعادل‌سازی بار معاصر در ادبیات بهتر عمل می‌کند. یافته‌ها نشان می‌دهند که رویکرد متعادل‌سازی بار پیشنهادی می‌تواند به طور موثر از منابع استفاده کند و در عین حال از درجه نسبتاً پایین عدم تعادل بار هنگام برخورد با سطوح مختلف ناهمگونی در یک محیط محاسباتی شبکه اطمینان حاصل کند[۸].

یک مدل متعادل کننده بار مبتنی بر عامل جدید برای بهبود عملکرد شبکه در مقاله‌ای ارائه داد. آنها معتقدند که در محاسبات شبکه، اگر بار سیستم در هر یک از منابع تقریباً برابر باشد، این نشان دهنده استفاده صحیح از منابع است. این مقاله یک مدل متعادل کننده بار مبتنی بر عامل را ارائه می-دهد. به جای متعادل کردن بار در شبکه با تکنیک مهاجرت شغلی یا انتقال کل فرآیند به منابع کم بار. مدل موازنه بار مبتنی بر عامل پیشنهادی با هدف بهره‌گیری از ویژگی‌های عامل برای تولید یک سیستم مستقل است. همچنین دارای مزیت متمایز نسبت به سایر مدل های متعادل کننده بار مبتنی بر عامل از نظر استفاده از منابع است. ارزیابی عملکرد نشان می‌دهد که مدل پیشنهادی می‌تواند عملکرد کلی محاسبات شبکه را افزایش دهد[۹].

یک مدل برای توازن بار خود بار شبکه: پارادایم فرآیند عامل در مقاله‌ای ارائه داد. آنها معتقدند که هدف متعادل سازی بار، بهره برداری عادلانه از منابع شبکه است، به گونه‌ای که هیچ منبعی بیش از حد بارگذاری نشود در حالی که سایر منابع کم بار شده یا بیکار هستند. رویکردهای زیادی پیشنهاد و اجرا شده است، اما با ظهور زیرساخت‌های جدید مانند شبکه‌ها و محاسبات جهانی، چالش‌های جدیدی با توجه به تأخیر شبکه مطرح می‌شود. خط مشی مکان‌یابی، به عنوان یکی از اصول اصلی راه حل های متعادل کننده بار، با هدف مکان‌یابی گره‌های اضافه بار و کم بار در یک شبکه است. برای انجام این کار، چندین پیام ارتباطی در سراسر شبکه ارسال می شود. این تکنیک منابع شبکه را هدر می‌دهد و باعث تاخیرهای قابل توجه شبکه در محیط‌هایی مانند محاسبات جهانی می‌شود که آن را غیرعملی می‌کند. در این مقاله، یک پارادایم جدید برای متعادل‌سازی بار توزیع‌شده تطبیقی با الهام از هوش ازدحام و سیستم‌های چند عاملی پیشنهاد شده است. در چنین پارادایمی، هیچ سرویس متعادل کننده بار مورد نیاز نخواهد بود. در واقع، وظایف کاری خودآگاه هستند و قادرند خود بارگذاری را در یک شبکه با بار ناشناخته متعادل کنند. به دلیل ماهیت خود و بر اساس مکانیسم های انگ، فرکانس ارتباط پارادایم پیشنهادی در مقایسه با راه حل‌های موجود به طور قابل توجهی کاهش می‌یابد. کار حاضر مبانی این پارادایم را که الگوریتم فرآیند عامل ابداع شده و همچنین الگوریتم های زیربنایی آن را توضیح می‌دهد. نتایج ارزیابی عملکرد در پایان این مقاله ارائه و مورد بحث قرار گرفته است[۱۰].

همچنین یک الگوریتم متعادل کننده بار جدید با کنترل کننده منطق فازی در محاسبات شبکه در یک مقاله ارائه شد. آنها معتقدند که تعادل بار نقش مهمی در محاسبات شبکه برای مدیریت مشاغل و تخصیص آنها به منابع کافی دارد. یک الگوریتم متعادل کننده بار کارآمد می‌تواند زمان اجرا را به حداقل برساند و با دستیابی به تعادل بار بین منابع در شبکه، استفاده از منابع را به حداکثر برساند. در این مقاله، یک الگوریتم موازنه بار مبتنی بر فازی ساده و کارآمد پیشنهاد شده است که نه تنها مدیریت منابع کارآمد را ارائه می‌دهد، بلکه استفاده بالا از منابع پویا را تضمین می‌کند. در این مقاله، طراحی و اجرای یک الگوریتم موازنه بار مبتنی بر منطق فازی پیشنهادی در محاسبات شبکه به تفصیل شرح داده شده است. الگوریتم پیشنهادی با استفاده از یک سیستم استنتاج منطق فازی کار می‌کند که از برخی معیارها برای گرفتن بار پردازنده و طول صف برای تعیین وضعیت هر گره استفاده می‌کند. سپس بر اساس حالت‌های کلی گره‌ها، وضعیت خوشه مربوطه برای تخصیص کارهای تازه وارد تعریف می‌شود تا تعادل بار بین خوشه‌ها و گره‌های مختلف انجام شود. عملکرد الگوریتم پیشنهادی بر اساس استفاده از منابع، زمان پاسخ و تأخیر ارزیابی می‌شود و با الگوریتم‌های زمان‌بندی مانند دریافت خدمات به

ترتیب ورود[1]، کوتاه‌ترین کار اول[2]، اولویت با زودترین موعد اول مقایسه می‌شود. الگوریتم مهاجرت شغلی برای تعادل بار پویا برای تجزیه و تحلیل عملکرد الگوریتم متعادل کننده بار مبتنی بر فازی، یک شبیه سازی با استفاده از دو کتابخانه جاوا منطق فازی و شبیه ساز گرید سیم[3] انجام شده است و نتایج به تفصیل ارائه شده است[۱۱].

۳ - تحلیل روش پیشنهادی

با توجه به پیشرفت در فناوری ارتباطات و استفاده از اینترنت، رشد بالقوه‌ای در فناوری محاسبات ابری وجود داشته است و مشکل در مقیاس بزرگ را حل می‌کند. رایانش ابری هم به سخت افزار و هم به برنامه به عنوان منابعی از طریق اینترنت برای کاربران اجازه می‌دهد. رایانش ابری مسیری به سوی محاسبات ابزاری است که در بسیاری از صنایع نرم افزاری چنگ می زند[۱۲]. ابر دارای سه مدل سرویس مانند نرم افزار به عنوان سرویس، پلتفرم به عنوان سرویس و زیرساخت به عنوان سرویس است و ابر دارای چهار مدل استقرار به نام های عمومی، خصوصی، ترکیبی و اجتماعی است.

تعادل بار یک ویژگی ضروری در تمام محیط‌های محاسباتی است، زیرا در سناریوی امروزی حجم زیادی از داده‌ها در فضای ابری ذخیره می‌شوند. این یک عملکرد حیاتی است که به ارائه کارایی بهتر کاربر ابر در استفاده از منابع از ابر کمک می کند. موازنه بار مورد استفاده توسط محققان نیاز و مزایای متعادل کردن بار در محیط محاسبات ابری را برجسته می کند. برخی از مزایای تعادل بار در زیر مورد بحث قرار گرفته است:

کاربرد با کارایی بالا: در مقایسه با روش های کلاسیک موجود، الگوریتم متعادل کننده بار پیاده سازی آسان و کم هزینه تر است. برنامه هایی که در سمت مشتری اجرا می شوند نیاز به عملکرد بهتری دارند و باید با هزینه بسیار پایین اجرا شوند. رویکرد متعادل‌سازی بار، عملکرد هر گره را بداهه می‌بخشد و از این رو عملکرد کلی سیستم بهبود می‌یابد و توان عملیاتی بهتر و قابلیت اطمینان افزایش می‌یابد. رویکرد متعادل کننده بار دارای مزیتی است که مسئله گرسنگی را حل می کند زیرا مشاغل کوچک نیازی به انتظار طولانی مدت ندارند.

افزایش مقیاس پذیری: ترافیک در ابر و مقیاس پذیری به طور موثر با رویکردهای متعادل کننده بار مناسب مدیریت می شود. با استفاده از متعادل کننده های بار موثر، ترافیک تقویت شده را می توان به طور یکنواخت در بین سرورهای مختلف توزیع کرد.

ترافیک افزایش یافته: سناریوهایی در دانشگاه زمانی که نتایج را برنامه‌ریزی می‌کنند، زمانی که سایت چندین درخواست را به طور همزمان دریافت می‌کند، دچار خرابی می‌شود. این می تواند به طور موثر با استفاده از رویکردهای متعادل کننده بار با توزیع ترافیک در سرورهای مختلف موجود کنترل شود. این رویکرد هر تعداد افزایش ناگهانی ترافیک را کنترل می کند.

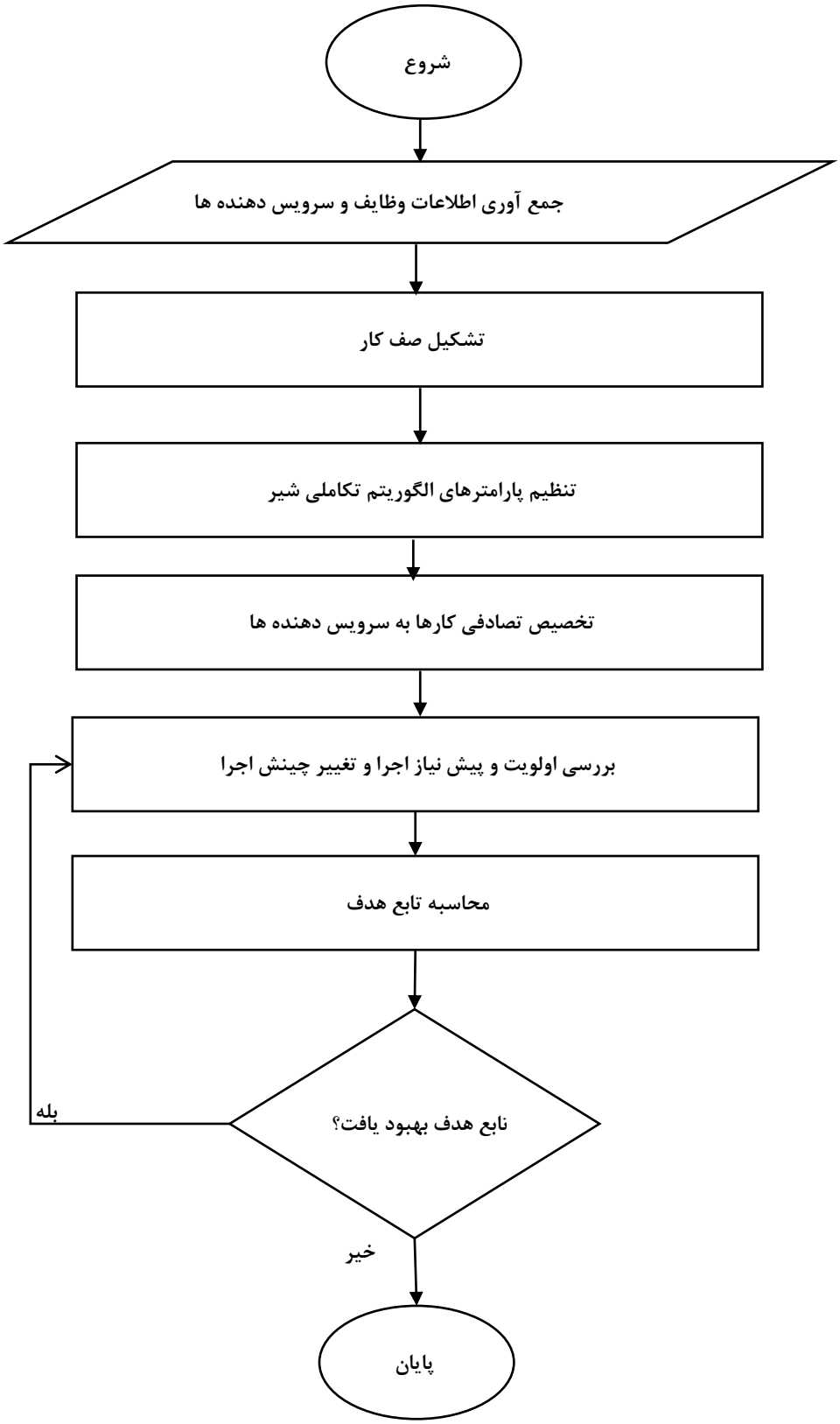
تداوم کسب و کار: روش متعادل سازی بار به جلوگیری از قطع شدن سریع صفحه وب کمک می کند. هر زمان که بار حتی در چندین سرور با وجود خرابی گره توزیع شود، کار می تواند به سایر گره های موجود منتقل شود. وجود رویکرد متعادل کننده بار انعطاف پذیری و مقیاس پذیری بهتری را در مدیریت ترافیک شبکه فراهم می کند.

الگوریتم فراابتکاری می تواند به طور موثر مسائل پیچیده را با کاوش و بهره برداری مکرر از فضای جستجو حل کند. مسئله تعادل بار در محیط محاسبات ابری یک مسئله بهینه سازی است که می تواند به طور موثر با استفاده از الگوریتم فراابتکاری مانند بهینه‌ساز شیر[4] حل شود و دارای چندین مزیت مانند:

حجم کار و تخصیص منابع در محیط ابری. الگوریتم‌های فراابتکاری می‌توانند با تغییرات دینامیکی سازگار شوند و در طول این نوع سناریوها راه‌حل‌های تقریباً بهینه را پیدا کنند.

تعادل بار شامل یافتن توزیع بهینه وظایف در چندین سرور است. الگوریتم‌های فراابتکاری از استراتژی‌های اکتشاف و بهره‌برداری در فضای جستجوی بزرگ برای یافتن راه‌حل بهینه استفاده می‌کنند.

[1] First Come First Serve (FCFS)
[2] Shortest Job First (SJF)
[3] GridSim
[4] Lion Optimization



شکل۱: فلوچارت روش پیشنهادی

الگوریتم های فراابتکاری را می توان بر اساس نیازهای خاص بسته به نیاز تعادل بار تنظیم کرد. الگوریتم های فراابتکاری قابل موازی سازی هستند که می توانند به طور موثر از مزایای قابلیت های محاسبات موازی استفاده کنند.

فلوچارت روش پیشنهادی در شکل نشان داده شده است. براساس شکل ۱ روش پیشنهادی بعد از جمع‌آوری اطلاعات کارهای یا سرویس گیرندگان درخواست کننده سرویس لیستی از وظایف با اولویت درخواست اجرا و پیوستگی و ارتباط بین آن‌ها تشکیل می‌دهد، این لیست شامل اطلاعات وظایف و منابع درخواستی از سرویس دهنده گرید است و این لیست سپس به روش پیشنهادی به عنوان ورودی داده می‌شود با توجه به ساختار الگوریتم فرابتکاری شیر ابتدا بررسی ورودی و تنظیم پارامترها انجام می‌شود و به صورت تصادفی کلیه وظایف یک خروجی اولیه دریافت می‌نمایند با توجه به موراد اولیه تابع هدف حداکثر سازی اجرا می‌شود و یک دور الگوریتم تخصیص صورت می‌پذیرد در صورتی که تابع از دور قبلی خود مقدار بیشتری نشان دهد الگوریتم یکبار دیگر اجرا می‌شود و در صورت برابری با دور قبلی الگوریتم خاتمه و لیست انتهایی اجرایی بدست آمده برای اجرا در اختیار سرویس‌دهندگان قرار خواهد گرفت.

یک الگوریتم الهام گرفته شده از بهینه‌ساز شیر برای جلوگیری از مشکل تعادل بار در گرید طراحی شده است. هدف این است که الگوریتم پیشنهادی در مقایسه با روش‌های انتخاب‌شده به‌عنوان معیار، نرخ اکتشاف و بهره‌برداری بهتری داشته باشد و همیشه در طول فرآیند جستجو برای دستیابی به بهینه جهانی تلاش می کند. در ادامه به تشریح این قسمت خواهیم پرداخت.

الگوریتم تکاملی بهینه سازی شیر برای تعادل بار در گرید: الگوریتم بهینه سازی شیر که یزدانی و جولای [۱۳] توسعه یافته اند، از رفتار زندگی و شکار شیرها الهام گرفته شده است. از آنجایی که این الگوریتم برای فضاهای جستجوی پیوسته پیشنهاد شده است، برخی تنظیمات باید برای اعمال بهینه‌ساز شیر در فضای جستجوی گسسته انجام شود. بنابراین، یک استراتژی نمایش راه حل کلید تصادفی در این الگوریتم به کار گرفته شده است. این روش پیشنهادی با تقلید از رفتار الگوریتم بهینه سازی شیر ایجاد شده است. شیرها متعلق به گونه های گربه وحشی هستند که همکاری و پرخاشگری بالایی را بین آنها از خود نشان می دهند. شیرها به دو دسته اجتماعی عشایر و ساکن طبقه بندی می شوند. ساکنان عموماً در گروهی به نام غرور زندگی می‌کنند. گروه غرور شامل پنج شیر ماده، توله و یک یا چند شیر نر است. شیرهای نر جوان معمولاً وقتی از نظر جنسی بالغ می شوند از غرور خود محروم می شوند. عشایر دسته دوم عموماً به تنهایی یا به صورت جفت زندگی می کنند. شیر ممکن است در طول چرخه زندگی خود بین غرور به عشایر جابه جا شود و عشایر به غرور را تشکیل دهد.

معیارهای ارزیابی: برای اندازه‌گیری کیفیت مجموعه راه‌حل غیرمسلط به‌دست‌آمده توسط یک الگوریتم مشخص، دو شاخص کیفیت یعنی حجم بیش از حد [۱] و اپسیلون یگانی[۲] استفاده می‌شود. که به عنوان منطبق با پارتو نشان داده شده اند و به عنوان شاخص های پیشرفته برای ارزیابی الگوریتم های چند هدفه در نظر گرفته می شوند. بسیاری از محققان از آنها برای ارزیابی عملکرد الگوریتم های زمان بندی چندهدفه استفاده کرده اند. علاوه بر این، با در نظر گرفتن دو شاخص کیفیت به جای یک، استحکام نتیجه گیری را افزایش می دهد. اگر این دو شاخص کیفیت برای دو الگوریتم نتایج متناقضی به دست آورند، به این معنی است که هیچ کدام را نمی‌توان برتر دانست.

حجم بیش از حد، مساحت فضای هدف تحت پوشش (مسلط) توسط یک جبهه تقریبی پارتو ارائه شده توسط یک الگوریتم معین را اندازه گیری می کند. با توجه به یک جبهه تقریبی پارتو P، هر مقدار هدف از هر راه حل $x \in P$ ابتدا باید با استفاده از نرمال سازی min-max در بازه [۰، ۱] نرمال شود. مقدار نرمال شده برای هدف i-امین محلول x (یعنی $fi'(x)$) را می توان به صورت زیر محاسبه کرد:

$$f_i'(x) = \frac{f_i(x) - min f_i}{max f_i - min f_i} \qquad (۱)$$

که در آن $\min fi$ و $\max fi$ به ترتیب مقادیر حداقل و حداکثر برای هدف i در تمام جبهه های پارتو (راه حل های غیر غالب) به دست آمده توسط همه الگوریتم های مقایسه شده در این مطالعه هستند. $fi(x)$ i-امین مقدار هدف راه حل است. سپس، اجازه دهید $zr = z1$، یک نقطه مرجع باشد که همه بردارهای هدف پارتو نرمال شده بر آن مسلط هستند. نقاط در P با ترتیب نزولی مقدار هدف اول مرتب شده اند. نشانگر IH

[1] Hypervolume (IH)

[2] Unary epsilon ($I\epsilon1$)

برای فضای هدف دو بعدی را می توان به صورت زیر محاسبه کرد:

$$I_H(P) = \prod_{k=1}^{|P|}(z_1^r - f_1'(x_k)) \times (f_2'(x_k) - f_2'(x_{k-1})) \quad (۲)$$

جایی که $f2\ '(xk-1)$ روی $z2\ r$ تنظیم می شود که ۱ = ۱. نقطه مرجع برای بستن هایپرحجم مورد نیاز است. با توجه به ادبیات نقطه مرجع باید با در نظر گرفتن بدترین مقدار برای هر هدف در کل مجموعه فونت های Pareto تولید شده توسط همه روش ها را برای یک نمونه خاص مقایسه کرده و آنها را در ۲۰٪ ضرب می کنیم. از آنجایی که مقادیر هدف بین [۰، ۱] نرمال می شوند، نقطه مرجع $zr = 1.2$، 1.2 تنظیم می شود و حداکثر مقدار IH 1.2 × 1.2 = 1.44 است. مقدار زیاد IH هم همگرایی بهتر و هم پوشش خوب جبهه پارتو بهینه (مرجع) را نشان می دهد.

اپسیلون یگانی، حداقل فاصله بین یک جبهه پارتو تقریبی داده شده و جبهه پارتو بهینه (ارجاع شده) را اندازه گیری می کند. برای یک جبهه پارتو تقریبی P و جبهه پارتو بهینه (ارجاع شده) R، هر مقدار هدف از راه حل $x \in P$ ابتدا باید در بازه [۱، ۲] نرمال شود. مقدار نرمال شده برای هدف i- امین راه حل x (یعنی $fi'(xk)$) را می توان به صورت زیر محاسبه کرد:

$$f_i'(x) = \frac{f_i(x) - \min f_i}{max f_i - min f_i} + 1 \quad (۳)$$

که در آن fi min، fi max و x (fi) با معادله یکسان هستند ۴-۳. سپس، نشانگر $I\epsilon\ 1$ را می توان به صورت ۴ محاسبه کرد.

$$I_\epsilon^1(P,R) = max_{y \in R} min_{x \in P} max_{1 \le i \le 2} \frac{f_i'(x)}{f_i'(y)} \quad (۴)$$

که در آن مقادیر هدف نرمال شده و به [۱، ۲] ترجمه می شوند تا از تقسیم بر صفر خطا در محاسبه شاخص جلوگیری شود. به این ترتیب، نشانگر بین ۱ و ۲ تغییر می کند. مقدار نزدیک به ۱ نشان می دهد که جبهه پارتو در نظر گرفته شده نزدیک به جبهه پارتو مرجع (بهینه) است، در حالی که مقدار انتخاب شده روی ۲ به معنای دور بودن آن است. از آنجایی که جبهه پارتو بهینه مسئله در نظر گرفته شده ناشناخته است، مجموعه اتحاد راه حل های غیر غالب به دست آمده توسط همه الگوریتم های آزمایش شده به عنوان جبهه پارتو مرجع (بهینه) انتخاب می شود. مقدار کوچکتر $I\epsilon\ 1$ به این معنی است که مجموعه پارتو تقریبی بدست آمده P به جلوی پارتو مرجع نزدیکتر است و کیفیت بهتری دارد.

توجه داشته باشید که ارزیابی فوری جبهه پارتو که پس از تکمیل هر اجرای الگوریتم به دست می آید، ضروری نیست. تنها زمانی که همه الگوریتم‌های مقایسه شده تمام مراحل خود را کامل کنند و تمام جبهه‌های پارتو تولید شده توسط آنها جمع‌آوری شود، می‌توانیم جبهه‌های پارتو به‌دست‌آمده توسط یک الگوریتم را ارزیابی کنیم. مقادیر حداقل و حداکثر (یعنی حداقل fi و حداکثر fi) هر هدف از این جبهه های پارتو تولید شده یافت می شود. با این روش، مقادیر حداقل و حداکثر برای هر راه حل غیر غالب ثابت و منصفانه است. مقادیر هدف هر راه حل غیر غالب را می توان به طور قانونی در فواصل مشخص شده ترسیم کرد.

۴- نتایج و ارزیابی روش پیشنهادی

این بخش به ارزیابی عملکرد روش پیشنهادی برای مشکل در نظر گرفته شده اختصاص دارد. در بخش‌های فرعی زیر، ابتدا تنظیمات آزمایشی نشان داده شده است که شامل نمونه‌های آزمایشی، معیارهای عملکرد و تنظیم پارامترها است. سپس سهم هر استراتژی بهبود الگوریتم پیشنهادی بررسی می‌شود. در نهایت، عملکرد روش پیشنهادی با مقایسه چندین الگوریتم پیشرفته ارزیابی می‌شود. علاوه بر این، لازم به ذکر است که الگوریتم های درگیر در ارزیابی توسط متلب کدگذاری شده و بر روی سروری با دو پردازنده Intel(R) Xeon(R) E5-2650 v3 @ 2.30GHz و G۱۲۸ رم تحت ویندوز ۷ اجرا می‌شوند.

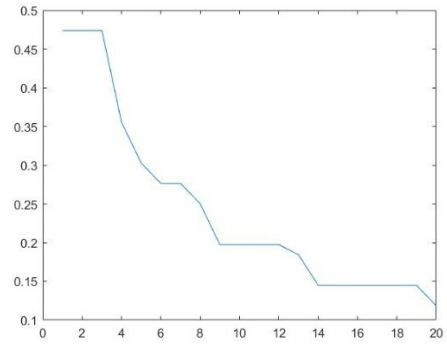
نتایج و ارزیابی روش پیشنهادی: در روش پیشنهادی از الگوریتم تکاملی بهینه‌سازی شیر استفاده شده است. این الگوریتم برای اجرا باید تعداد تکرار مشخصی را برای اتمام به الگوریتم در نظر بگیرد این تعداد باید عددی مناسبی باشد. زیرا این تعداد در صورت زیاد بودن زمان اجرای روش را غیرمنطقی می‌نماید و در صورتی که این تعداد بسیار کم باشد به مقدار بهینه مورد نظر دست نیافته و نتایج نهایی مناسب نخواهند بود. در شکل ۲ نتایج با تعداد تکرار مختلف در این الگوریتم نشان داده شده است.

همانگونه که در شکل ۲ نشان داده شده است مقادیر در تعداد ۴۰ عدد به همگرایی لازم می‌رسد و نیاز به تکرار بیشتر نیست و یا این تعداد می‌توان به بهینه سازی لازم دست یافت بنابراین براساس این شکل‌ها مقدار ۴۰ را به عنوان عدد مناسب تعداد تکرار برمی‌گزینیم.

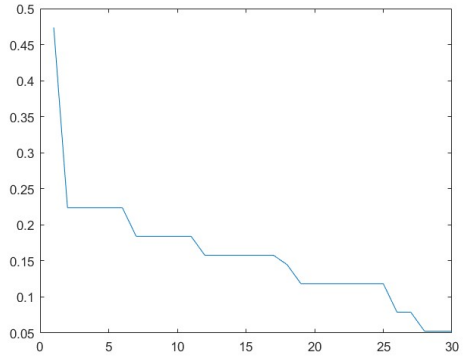
مجموعه داده: به منظور ارزیابی عملکرد الگوریتم-پیشنهادی و اجرای مقایسه با سایر الگوریتم‌های پیشرفته، از مجموعه معیار پیشنهادی استفاده شده که در http://people.brunel.ac موجود است. این مجموعه بنچمارک از ۱۲۰ نمونه تشکیل شده است که در ۱۲ گروه با ۱۰ نمونه سازماندهی شده است. هر گروه شامل ترکیب های مختلفی از تعداد کار n و تعداد ماشین m می باشد. ترکیبات{۲۰، ۵۰،۱۰۰}، {۵،۱۰،۲۰}، {۱۰،۲۰} و ۵۰۰ × ۲۰ هستند. برای سررسید هر شغل، چهار سناریو طراحی شده است. تاریخ های سررسید با توزیع یکنواخت تصادفی که در آن P یک کران پایین است، T ضریب تاخیر محدوده سررسید است. چهار سناریو در تمام تاریخ های سررسید ارزیابی موجود است.

این بخش به ارزیابی عملکرد روش پیشنهادی برای مشکل در نظر گرفته شده اختصاص دارد. در بخش‌های فرعی زیر، ابتدا تنظیمات آزمایشی نشان داده شده است که شامل نمونه‌های آزمایشی، معیارهای عملکرد و تنظیم پارامترها است. سپس سهم هر استراتژی بهبود الگوریتم پیشنهادی بررسی می‌شود. در نهایت، عملکرد روش پیشنهادی با مقایسه چندین الگوریتم پیشرفته ارزیابی می‌شود. علاوه بر این، لازم به ذکر است که الگوریتم های درگیر در ارزیابی توسط متلب کدگذاری شده و بر روی سروری با دو پردازنده Intel(R) Xeon(R) E5-2650 v3 @ 2.30GHz و G۱۲۸ رم تحت ویندوز ۷ اجرا می‌شوند.

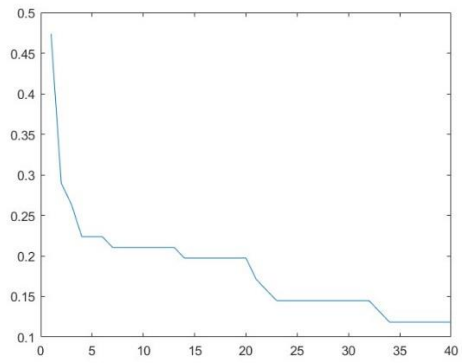
نتایج و ارزیابی روش پیشنهادی: در روش پیشنهادی از الگوریتم تکاملی بهینه‌سازی شیر استفاده شده است. این الگوریتم برای اجرا باید تعداد تکرار مشخصی را برای اتمام به الگوریتم در نظر بگیرد این تعداد باید عددی مناسبی باشد. زیرا این تعداد در صورت زیاد بودن زمان اجرای روش را غیرمنطقی می‌نماید و در صورتی که این تعداد بسیار کم باشد به مقدار بهینه مورد نظر دست نیافته و نتایج نهایی مناسب نخواهند بود. در شکل ۳ نتایج با تعداد تکرار مختلف در این الگوریتم نشان داده شده است ومقادیر در تعداد ۴۰ عدد به همگرایی لازم می‌رسد و نیاز به تکرار بیشتر نیست و یا این تعداد می‌توان به بهینه سازی لازم دست یافت بنابراین براساس این شکل‌ها مقدار ۴۰ را به عنوان عدد مناسب تعداد تکرار برمی‌گزینیم.



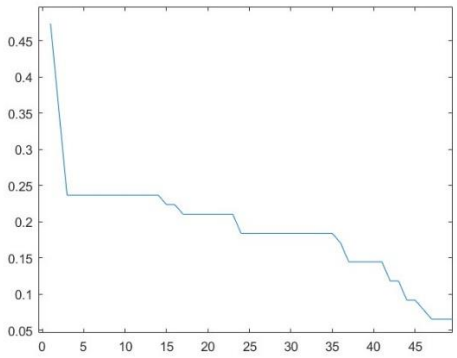
الف) با تعداد ۲۰ تکرار



ب) با تعداد ۳۰ تکرار



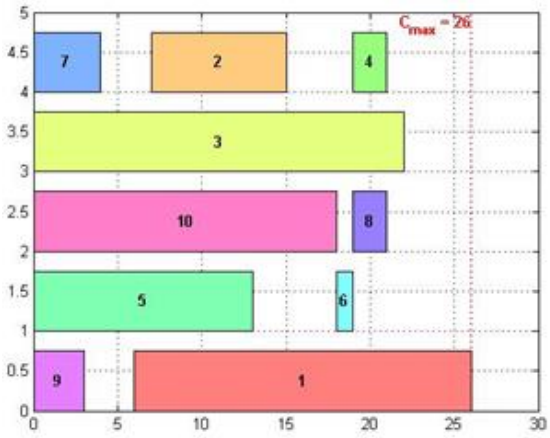
ج) با تعداد ۴۰ تکرار



د) با تعداد ۵۰ تکرار

شکل ۳: مقادیر تابع هزینه الگوریتم تکاملی بهینه سازی شیر با تعداد تکرار مختلف

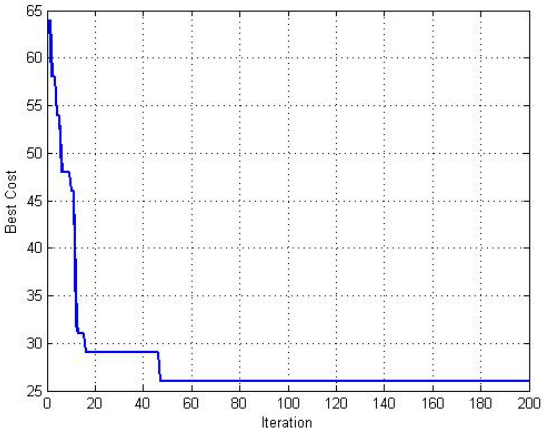
در گام بعدی کل الگوریتم را اجرا نموده و برای اجرای آن از یک محیط با ۵ ماشین مجازی مختلف با توانایی یکسان استفاده می‌شود. برای اجرا ده وظیفه با زمان‌های بارگذاری شده از مجموعه داده و به صورت تصادفی و با وابستگی‌های وظایف تولید شده که در روش پیشنهادی مورد استفاده قرار می‌گیرد، استفاده شده است. نتیجه اجرا این سناریو بر روی روش پیشنهادی در شکل ۴ نشان داده شده است. همانگونه که در شکل ۴ نشان داده شده است زمان کامل شدن کلیه کارها ۲۶ میلی ثانیه است. و طبق این خروجی وظایف ۱ و ۹ برروی ماشین مجازی ۱، وظایف ۵ و ۶ برروی ماشین مجازی ۲، وظایف ۸ و ۱۰ برروی ماشین مجازی ۳، وظیفه ۳ برروی ماشین مجازی ۴ و وظایف ۲،۴ و ۷ برروی ماشین مجازی شماره ۵ اجرا می‌شود. ترتیب اجرا وظایف از سمت چپ بوده و دلیل این ترتیب اولویت و وابستگی کارهایی است که در نظر گرفته شده است.



شکل ۴: زمانبندی ده وظیفه برروی پنج ماشین مجازی توسط روش پیشنهادی

در تکرارهای مختلف و استفاده از الگوریتم بهینه‌سازی تابع هزینه‌ای تولید می‌شود که تابع هزینه برای این مسئله در شکل۵ نشان داده شده است.

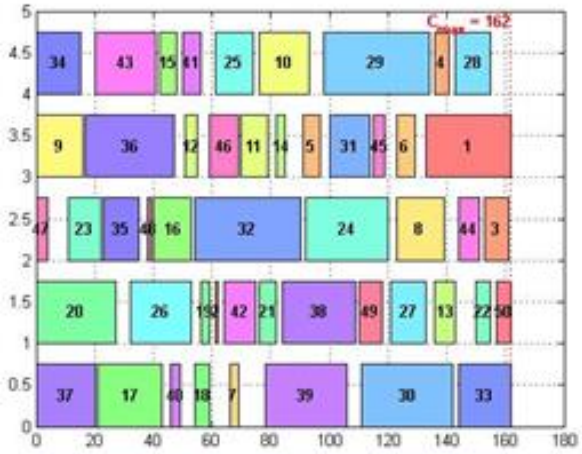
همانگونه که در شکل ۵ نشان داده شده است هزینه ابتدایی اولیه تولید شده برای روش پیشنهادی ۶۵ بوده است که این مقدار به مرور زمان کم و کمتر می‌شود تا به نقطه ۲۶ می‌رسد. در انتهای تکرارهای در نظر گرفته شده نتیجه این اجرا با مقدار تولید شده برای زمان اتمام کلیه کارها برابر است.



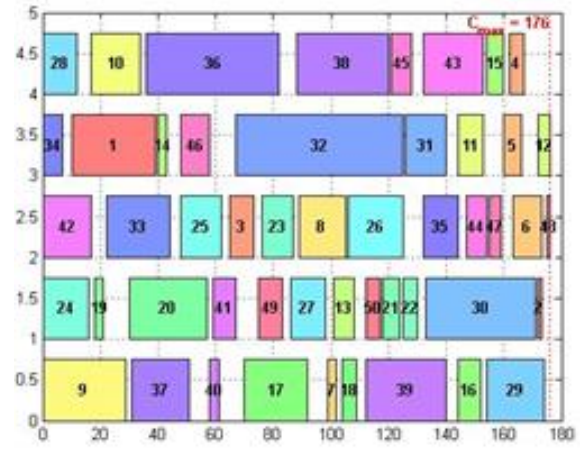
شکل ۵ تابع هزینه اجرای روش پیشنهادی برروی پنج ماشین مجازی و ده وظیفه

شکل ۵ اطلاعات جمع آوری شده توسط چهار روش را با اطلاعات اندازه گیری شده برای استفاده از پردازنده در زمانی که کد مشتری ۳ بود مقایسه می کند. فرکانس نوسان مورد انتظار مدل رگرسیون اوج بیزی برای عملکرد پردازنده بین ۰.۴۲ و ۰.۲۳ بود. فرکانس نوسان مورد انتظار مدل درخت تصمیم برای استفاده از پردازنده ۰.۳۲ تا ۰.۳۵ است. حد خطای پیش‌بینی مدل پشتیبان رگرسیون برداری گاوس برای استفاده از پردازنده بین ۰.۵۳ و ۰.۲۵ بود. مدل‌های رگرسیون بردار پشتیبان گاوس بهینه‌سازی شده با محاسبه مقیاس‌پذیر دارای طیف خطای پیش‌بینی مصرف پردازنده ۰.۲۲ تا ۰.۲۴ بودند. حاشیه پیش‌بینی استفاده از پردازنده سیستم پیشنهادی یا یک خطا کمترین حد از چهار استراتژی بوده است.

گام بعدی مقایسه روش پیشنهادی با یکی از روش‌های همسان بوده که برای این امر روش زمانبندی چندهدفه با استفاده از الگوریتم پایه [۱۴] در نظر گرفته شده است. برای این امر سه آزمایش مختلف با تعداد ماشین‌های مجازی ۵،۱۰ و ۲۰ انجام داده که در هر یک از آزمایش‌ها تعداد مختلفی وظیفه انتخاب شده و تولید شده به ماشین‌های مجازی جهت زمانبندی داده شده و از نظر زمان اتمام کار الگوریتم‌ها با یکدیگر مورد مقایسه قرار خواهند گرفت همچنین در این میان تابع هزینه دو الگوریتم نیز مقایسه می‌شود. نمونه‌ای از اجرای دو الگوریتم در شکل ۶ نشان داده شده است.



الف) روش پیشنهادی



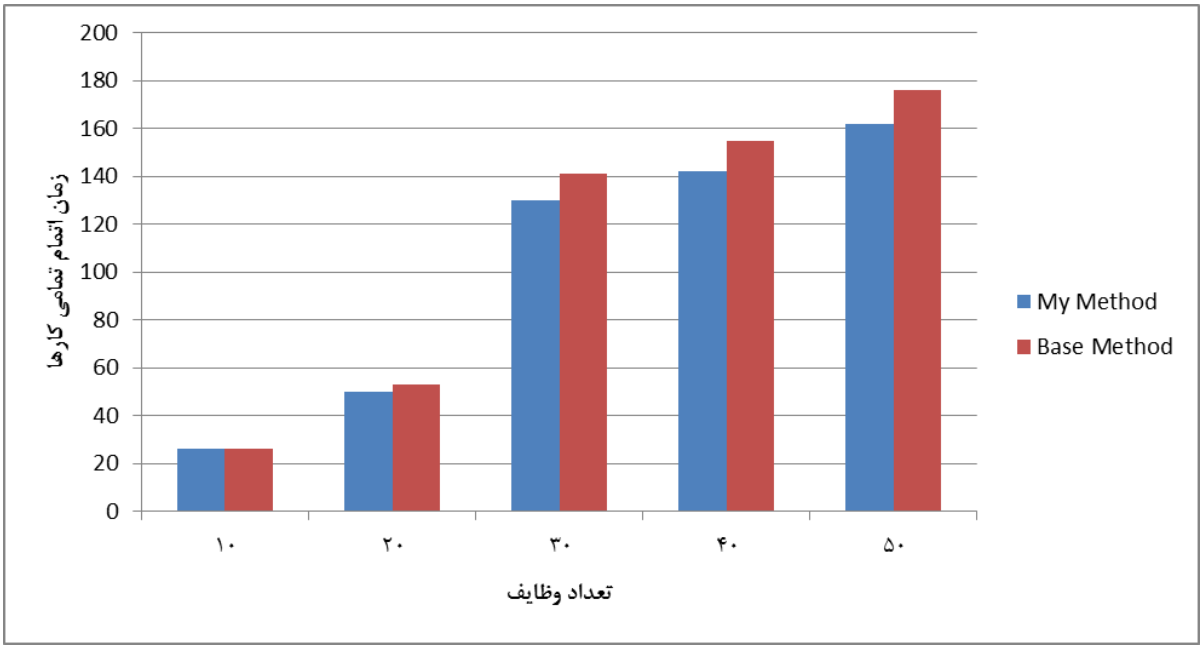
ب) روش پایه

شکل۶: نتیجه اجرای روش‌های مورد مقایسه با پنج ماشین مجازی و پنجاه وظیفه

همانگونه که در شکل ۶ نشان داده شده است روش پیشنهادی اجرای ۵۰ وظیفه در ۵ ماشین مجازی را با ۱۶۲ میلی ثانیه به اتمام رسانده که ترتیب اجرای وظایف در ماشین‌های مجازی مختلف نیز در شکل مشخص است ولی روش پایه همان تعداد وظایف و ماشین مجازی با زمان‌ها و اولویت‌های برابر را در ۱۷۶ میلی ثانیه به اتمام رسانده است.

نمودار اثرات میانگین تمام پارامترها در شکل ۶ نشان داده شده است. از این شکل می توان دریافت که عملکرد الگوریتم با افزایش Smax کاهش می یابد. به خصوص، عملکرد الگوریتم در Smax = ۷ و Smax = ۱۰ حداکثر تغییرات زیادی دارد. هنگامی که Smax = 20، الگوریتم بدترین عملکرد را دارد، در حالی که عملکرد با تنظیم ۳ = max به دست می آید. مرحله انطباق، و توانایی الگوریتم برای انجام تکرارهای بیشتر تحت تأثیر قرار خواهد گرفت. در واقع، وقتی Smax را روی ۱۰، ۱۵ و ۲۰ آزمایش می کنیم، تعداد تکرارهای الگوریتم بسیار محدود است، به خصوص برای نمونه های بزرگ، بنابراین نتایج خروجی بسیار ضعیف هستند. بنابراین، Smax باید به عنوان مقدار کوچک تنظیم شود، یعنی Smax = 3. دومین عامل مهم Pmax است. در شکل ۶ می توان مشاهده کرد که Pmax = 10 بهترین نتایج را به دست می آورد، در حالی که Pmax = 20 بدترین عملکرد را دارد. معمولاً، جمعیت زیادی می‌توانند درخواست کنند که یک جبهه پارتو با توزیع خوب به دست آید، اما هزینه محاسباتی را افزایش می‌دهد و سرعت همگرایی را کاهش می‌دهد به طوری که همگرایی تقریبی جبهه پارتو به‌دست‌آمده رضایت‌بخش نیست. Pحداکثر باید با در نظر گرفتن مبادله بین هزینه محاسباتی، همگرایی و توزیع ایجاد شود. پارامتر Smin رتبه سوم را به خود اختصاص می دهد. همانطور که در شکل ۶ مشاهده می شود، وقتی Smin برابر با ۰ تنظیم می شود، الگوریتم بهتر عمل می کند، که نشان می دهد ریشه دونده نسبتا ضعیف نباید حق تولید بذر خود را داشته باشند. آخرین عامل مهم σmax است که با ترکیب σmin تعادل بین اکتشاف و بهره برداری را کنترل می کند. σmin و σmax بزرگ، اکتشاف مورد نیاز برای جستجوی جهانی را تقویت می کند، در حالی که بهره برداری را که برای همگرایی خوب لازم است، کاهش می دهد. شکل ۶ نشان می دهد که الگوریتم زمانی عملکرد خوبی دارد که σmax برابر با ۳.۰ باشد، که نتیجه بالا را تأیید می کند.

همانگونه که در شکل ۶ نشان داده شده است تابع هزینه روش پایه در ابتدا بالاتر از روش پیشنهادی بوده ولی در انتها این تابع هزینه کمتر بوده و مقدار بهتری را نشان می‌دهد البته دلیل این امر استفاده همزمان از دو تابع بهینه‌سازی در روش پیشنهادی بوده که مقدار تابع هزینه را در انتها افزایش می‌دهد. هدف از ارایه روش پیشنهادی کاهش مقدار زمان اتمام کار بوده که مطابق شکل ۶ به این امر دست یافته‌ایم. نتایج مقایسه دو روش با تعداد وظایف مختلف در شکل ۷ نشان داده شده است.

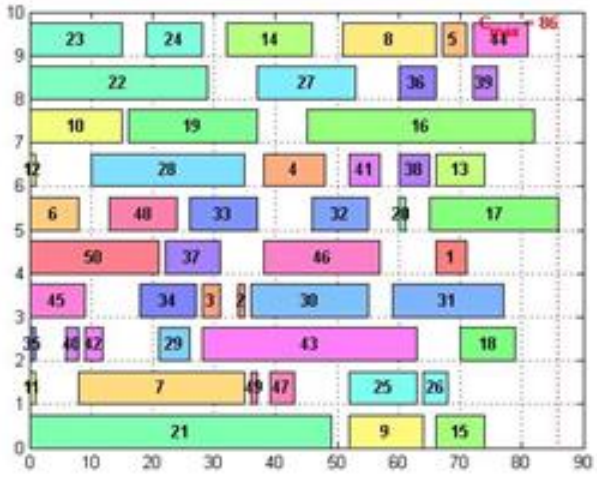


شکل۷: مقایسه روش پیشنهادی و روش پایه با تعداد وظایف مختلف بر روی ۵ ماشین مجازی

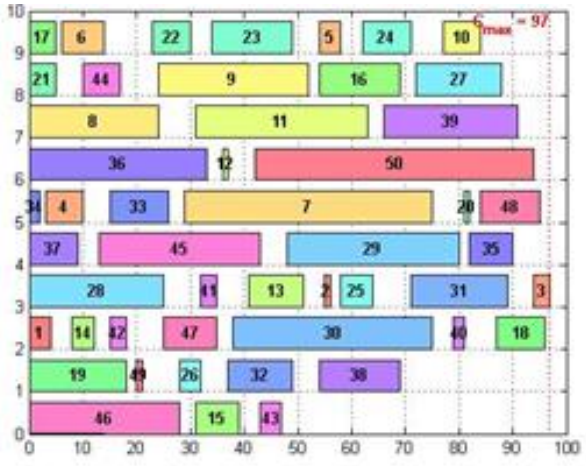
همانگونه که در شکل ۷ نشان داده شده است هر دو روش تعداد وظایف ۱۰ را با زمان یکسانی در ۵ ماشین مجازی به اتمام رسانده‌اند ولی زمانی که تعداد وظایف افزایش پیدا می‌کند روش پیشنهادی زمان اتمام کار پایین‌تری را از خود نشان می‌دهد و این بهبود با افزایش تعداد وظایف بیشتر می‌شود. اختلاف دو روش در تعداد ۳۰ وظیفه ۱۱ میلی ثانیه در ۴۰ وظیفه ۱۳ و در ۵۰ وظیفه به ۱۴ میلی ثانیه رسیده است.

شکل ۷ نموداری از تفاوت بین مصرف واقعی پردازنده ۴ روش و مصرف مورد انتظار پردازنده را در زمانی که شناسه سرور ۳ بود نشان می دهد. باید از نمودار مشخص شده باشد که کمترین مقدار نمونه اولیه و تصمیم رگرسیون رگرسیون بیزی مدل‌سازی درختی نزدیک به ۰ بوده است، بالاترین مقدار مشابه ۰.۲ بوده است، در حالی که فقط چند استثنای نادر وجود دارد، که نشان می‌دهد از دست دادن اعتبار دو روش ناچیز بوده و همچنین تثبیت بیشتر بوده است. اگرچه می‌تواند استفاده از پردازنده را با دقت زیادی پیش‌بینی کند، اما به دلیل موارد پرت زیاد، انسجام ندارد.

در ارزیابی بعدی تعداد ماشین‌های مجازی به ۱۰ عدد افزایش داده شده است. نمونه‌ای از اجرای دو الگوریتم در شکل ۸ نشان داده شده است.



الف) روش پیشنهادی

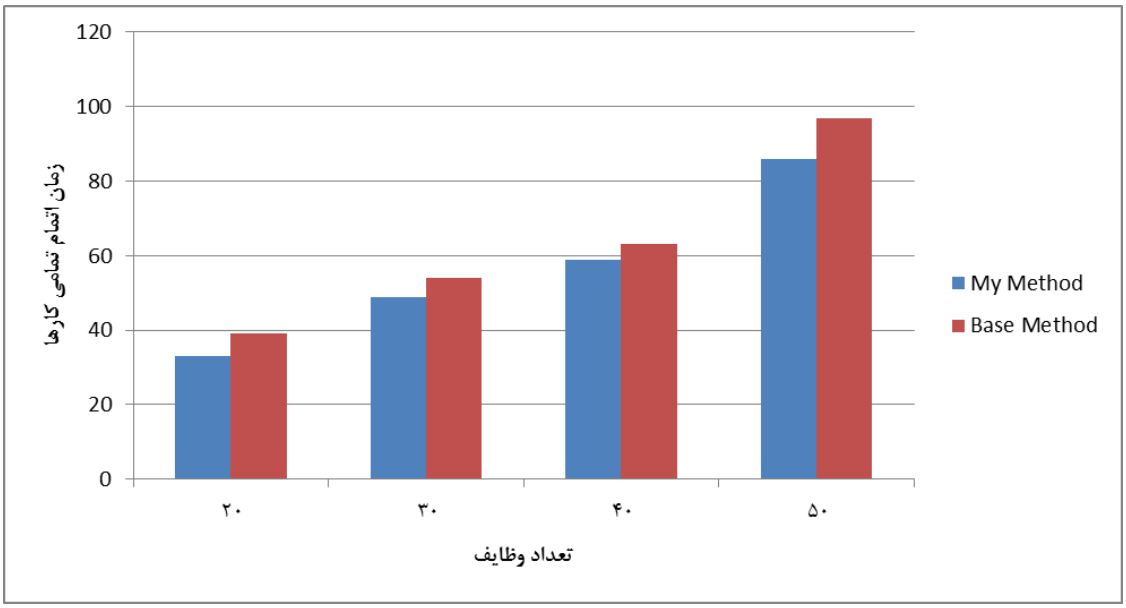


ب) روش پایه

شکل ۸: نتیجه اجرای روش‌های مورد مقایسه با ده ماشین مجازی و پنجاه وظیفه

همانگونه که در شکل ۸ نشان داده شده است روش پیشنهادی اجرای ۵۰ وظیفه در ۱۰ ماشین مجازی را با ۸۶ میلی ثانیه به اتمام رسانده که ترتیب اجرای وظایف در ماشین‌های مجازی مختلف نیز در شکل مشخص است (به عنوان مثال وظایف ۲۲،۲۷،۳۶ و ۳۹ برروی ماشین مجازی ۹ اجرا می‌شود) ولی روش پایه همان تعداد وظایف و ماشین مجازی با زمان‌ها و اولویت‌های برابر را در ۹۷ میلی ثانیه به اتمام رسانده است.

همانگونه که در شکل ۹ نشان داده شده است تابع هزینه روش پایه در ابتدا پایین‌تر از روش پیشنهادی بوده ولی در برخی از نواحی میانی عملیات بالاتر بوده است. در انتها نیز این تابع هزینه کمتر بوده و مقدار بهتری را نشان می‌دهد البته دلیل این امر استفاده همزمان از دو تابع بهینه‌سازی در روش پیشنهادی بوده که مقدار تابع هزینه را در انتها افزایش می‌دهد. هدف از ارایه روش پیشنهادی کاهش مقدار زمان اتمام کار بوده که این امر



شکل۹: مقایسه روش پیشنهادی و روش پایه با تعداد وظایف مختلف بر روی ده ماشین مجازی

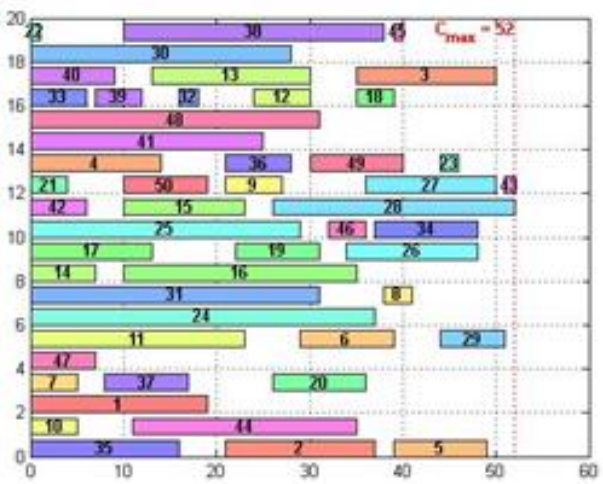
همانگونه که در شکل ۹ نشان داده شده است اختلاف در تعداد وظایف پایین کمتر بوده ولی زمانی که تعداد وظایف افزایش پیدا می‌کند روش پیشنهادی زمان اتمام کار پایین‌تری را از خود نشان می‌دهد و این بهبود با افزایش تعداد وظایف بیشتر می‌شود. اختلاف دو روش در تعداد ۳۰ وظیفه ۵ میلی ثانیه در ۴۰ وظیفه ۴ و در ۵۰ وظیفه به ۱۱ میلی ثانیه رسیده است.

دست یافته‌ایم. نتایج مقایسه دو روش با تعداد وظایف مختلف نشان داده شده است.

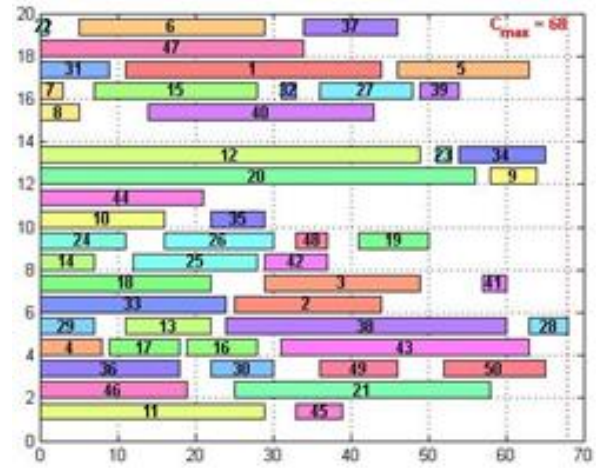
علاوه بر این، شکل ۹ همچنین تعاملات ۲ سطحی بین پارامترهای در نظر گرفته شده را نشان می دهد. از این شکل می بینیم که تعاملات $Pmax * Smax$، $Smin * Smax$ و $Smax * \sigma max$ از نظر آماری معنی دار هستند. شکل ۴-۵ نمودار تعاملات آنها را بر روی هر دو شاخص عملکرد نشان می دهد، جایی که می توان مشاهده کرد که این تعاملات ضعیف هستند و برخلاف نتایج بدست آمده از شکل ۹ نیستند. به خصوص، ما همچنین می‌توانیم ببینیم که تغییرات تعاملات $Smin * Smax$ و $Smax * \sigma max$ برای $Smin$ و σmax بسیار کوچک است، که بیشتر نشان می‌دهد که $Smin$ و σmax تأثیر بسیار کمی بر روی max دارد. بنابراین، با توجه به آزمایش پارامتری و تجزیه و تحلیل فوق، پارامترهای نهایی روش پیشنهادی به صورت زیر پیشنهاد می شود: $Pmax = 10$، $Smin = 0$، $Smax = 3$، $\sigma min = 0.35$ و $\sigma max = 3.0$.

همانطور که در شکل ۱۰ مشاهده می شود، مرکز خوشه با تکنیک انتخاب تصادفی انتخاب شد تا نمونه های داده های بنفش را در نیمه های بالایی و پایینی گروه اول جدا کند. خیلی بالا و پایین بود. محل نمونه برداری در بیضی قرمز و گروه پنجم فاصله چندانی نداشت. چند مشکل در طبقه بندی نقاط در نمونه بنفش با تکنیک کاملا تصادفی وجود دارد.

همانطور که در شکل۱۰الف مشاهده می شود، مرکز گروه بندی که روش min-max برای تقسیم نمونه های داده در بیضی قرمز به ۲ گروه انتخاب کرد، باید به دلیل نزدیکی نقاط داده به یک دسته خاص تقسیم می شد. با جداسازی این مکان‌ها، الگوریتم‌های min-max با مشکلات خاصی مواجه شدند. مطابق شکل ۱۰ ب، روش حداقل - حداکثر اصلاح شده ارائه شده در روش پیشنهادی با تقسیم مراکز تمرکز اولیه بهتر سازگار بود. در نتیجه، سیستم خوشه‌بندی اولیه برای رویکرد k-average مورد استفاده در این تحقیق، رویکرد Min-max بهینه‌سازی شده بود.



الف) روش پیشنهادی



ب) روش پایه

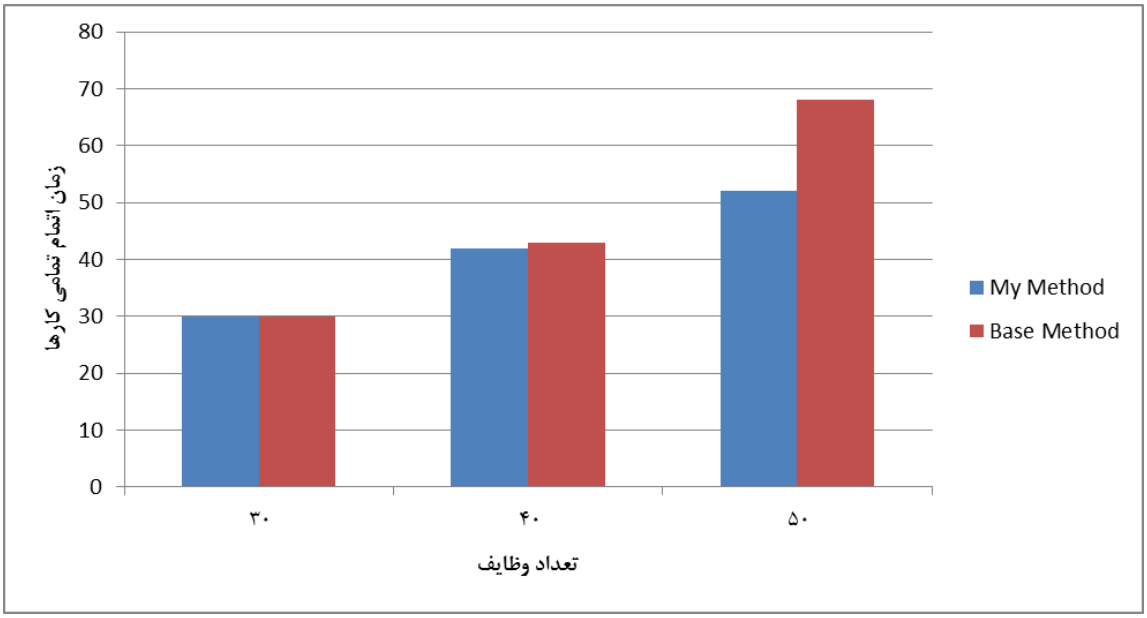
شکل ۱۰ نتیجه اجرای روش‌های مورد مقایسه با بیست ماشین مجازی و پنجاه وظیفه

همانگونه که در شکل ۱۰ نشان داده شده است روش پیشنهادی اجرای ۵۰ وظیفه در ۲۰ ماشین مجازی را با ۵۲ میلی ثانیه به اتمام رسانده که ترتیب اجرای وظایف در ماشین‌های مجازی مختلف نیز در شکل مشخص است (به عنوان مثال وظایف ۱۷،۱۹ و ۲۶ برروی ماشین مجازی ۱۰ اجرا می‌شود) ولی روش پایه همان تعداد وظایف و ماشین مجازی با زمان‌ها و اولویت‌های برابر را در ۶۸ میلی ثانیه به اتمام رسانده است.

همانگونه که در شکل ۱۱ نشان داده شده است تابع هزینه روش پایه در ابتدا بالاتر از روش پیشنهادی بوده ولی در انتها این تابع هزینه کمتر بوده و مقدار بهتری را نشان می‌دهد البته دلیل این امر استفاده همزمان از دو تابع بهینه‌سازی در روش پیشنهادی بوده که مقدار تابع هزینه را در انتها افزایش می‌دهد. هدف از ارایه روش پیشنهادی کاهش مقدار زمان اتمام کار بوده که مطابق شکل ۱۰به این امر دست یافته‌ایم. نتایج مقایسه دو روش با تعداد وظایف مختلف در شکل ۱۱ نشان داده شده است.

همانگونه که در شکل ۱۱ نشان داده شده است اختلاف در تعداد وظایف پایین کمتر بوده حتی در تعداد ۳۰ وظیفه هر دو روش در ۳۰ میلی ثانیه تمامی کارها را به پایان رسانده‌اند ولی زمانی که تعداد وظایف افزایش پیدا می‌کند روش پیشنهادی زمان اتمام کار پایین‌تری را از خود نشان می‌دهد و این بهبود با افزایش تعداد وظایف بیشتر می‌شود. اختلاف دو روش در تعداد ۴۰ وظیفه ۱ میلی ثانیه و در ۵۰ وظیفه به ۱۶ میلی ثانیه رسیده است. روش پیشنهادی به یک مشکل زمان‌بندی تک ماشینی می‌پردازد که در آن ماشین باید پس از یک بازه دوره‌ای ثابت نگهداری شود. هدف این است که کل انحراف مطلق زمان های تکمیل را برای مشکل پرداخته شده به حداقل برسانیم. نشان داده شد که مشکل به معنای قوی NP-hard است. بنابراین، برای مقابله با این مشکل NP-hard، یک الگوریتم فراابتکاری جدید، یعنی الگوریتم بهینه‌سازی شیر، پیشنهاد شد. این الگوریتم رفتار فردی و اجتماعی شیرها را برای بهینه سازی تقلید می کند. به منظور ارزیابی کارایی الگوریتم بهینه‌سازی شیر پیشنهادی، با برخی از الگوریتم‌های فراابتکاری جدید شناخته شده مقایسه شد. نتایج مقایسه برتری مطلق الگوریتم پیشنهادی ما را نشان داد. یکی از محدودیت های تحقیق ما این است که پارامترها قطعی در نظر گرفته می شوند. رفع این محدودیت باعث می شود مشکل به شرایط واقعی نزدیک شود. علاوه بر این، در نظر گرفتن سایر توابع هدف می تواند یک جهت تحقیقات آینده جالب باشد. همچنین تکنیک های دیگری مانند فراابتکاری ترکیبی،

اکتشافی، بهینه سازی قوی و برنامه ریزی تصادفی قابل مطالعه و مقایسه با روش پیشنهادی است. علاوه بر این، مشکل را می توان در یک محیط ماشینی پیچیده تر مانند محیط کارگاه بررسی کرد، به ویژه زمانی که برخی از مفروضات اضافی مانند پردازش وابسته به ترتیب و زمان های تنظیم یا اثرات یادگیری درگیر هستند..



شکل ۱۱: مقایسه روش پیشنهادی و روش پایه با تعداد وظایف مختلف بر روی ده ماشین مجازی

۶- نتیجه گیری

این مطالعه مروری بر تامین منابع محاسبات ابری و الگوریتم‌های زمان‌بندی کار، و همچنین بررسی طبقه‌بندی روش‌ها را پوشش می‌دهد. با استفاده از الگوریتم‌های زمان‌بندی برای انتخاب مناسب‌ترین منبع، زمان‌بندی کار برای بهبود ویژگی‌های تعیین‌کننده عملکرد کلیدی مانند زمان پاسخ، زمان ساخت، زمان جریان، زمان پایان، هزینه و استفاده از منابع طراحی شده است. ما چند تکنیک زمان‌بندی پیشرفته را بررسی کردیم و آنها را بر اساس مشکلی که برای حل آن طراحی شده بودند طبقه‌بندی کردیم. در این تحقیق، مفاهیم اساسی و مزایای رویکردهای تامین منابع موجود، و همچنین طبقه‌بندی ریتم‌های الگوریتم زمان‌بندی به دسته‌های ایستا و پویا را پوشش داده‌ایم. الگوریتم‌های زمان‌بندی مهم دیگری مورد بحث قرار می‌گیرند، از جمله الگوریتم‌هایی که به رویکردهای اکتشافی و فراابتکاری و همچنین رویکردهای ترکیبی و مبتنی بر آموزش وابسته هستند. در این بخش، همه الگوریتم‌های ذکر شده در بالا را از نظر پارامترهای QoS، ماهیت وظایف، رویکردهای حل مسئله، مزایای الگوریتم مورد استفاده، محدودیت‌های الگوریتم مورد استفاده و ابزار شبیه‌سازی بررسی می‌کنیم، همگی در حالی که یک ضرب‌الاجل را به عنوان یک محدودیت در نظر می‌گیریم. . ما کشف کرده‌ایم که اکثر الگوریتم‌ها چندین فاکتور حیاتی کیفیت خدمات، محدودیت‌ها و نقض موافقت‌نامه سطح خدمات را از جمله موارد دیگر در نظر نمی‌گیرند. حداکثر الگوریتم ها شامل محدودیت های مختلفی هستند که باعث می شوند عملکرد الگوریتم در طول زمان کاهش یابد. هنگامی که صحبت از رایانش ابری می شود، یک چارچوب سیستم تولید به هنگام[1] یک راه حل جامع است که با ادغام چندین رویکرد مختلف در این زمینه مطالعاتی ارائه می شود. چارچوب پیشنهادی برای بهبود عملکرد زمان‌بندی کار موثر در حالی که به طور همزمان هزینه محاسباتی فرآیند را کاهش می‌دهد در نظر گرفته شده است. در رایانش ابری، مهم ترین چیزی که باید به خاطر داشت این است که زمان بندی کار باید منصفانه باشد. در روش پیشنهادی، ما یک تکنیک زمان‌بندی کار مبتنی بر اولویت را برای رایانش ابری ارائه می‌دهیم، که تعیین می‌کند کدام شغل بیشترین نتایج را ایجاد می‌کند. برای این مطالعه، تعدادی از مسائل به منظور ارزیابی ابتکارات روش‌شناسی‌های مختلف مدرن و چندین دستورالعمل مرتبط بررسی شد. انتظار می‌رود که روش پیشنهادی ما به عنوان پایه‌ای برای تحقیقات

[1] Just in time

بعدی در مورد برنامه‌ریزی موفقیت‌آمیز کار در عمق بیشتر عمل کند.

یک چالش کلیدی بهینه سازی سرعت شبکه با حفظ بی طرفی بود. راه حل‌های نظارت بر قدرت برای کاهش این عدم تعادل بار توسعه داده شده است. این تکنیک‌ها اغلب به سخت‌افزار یا نرم‌افزار منحصربه‌فرد برای شرکت‌کنندگان نیاز دارند. برای انجام سیستم محاسبات ابری تعادل بار، این مطالعه یک طرح برنامه‌نویسی ماشین مجازی هوشمند با استفاده از تکنیک یادگیری ماشین ایجاد کرد. محققان دو روش را معرفی کردند که پس از انجام یک بررسی عمیق، راه‌حل‌های احتمالی بهینه را برای موضوع شناسایی می‌کنند. یک راه حل متعادل کننده حداقل بار با روش دوم ایجاد می شود، در حالی که رویکرد اول بار روی نقاط دسترسی شلوغ شبکه را کاهش می دهد. بیایید به طور خاص بر روی نگاشت پیوندها تمرکز کنیم زیرا نقشه برداری گره ها از قبل ایجاد شده است. این مطالعه یک بهینه‌سازی شیر بهبود یافته نوآورانه و موثر با الگوریتم Min-Max در سیستم‌های واقعی ارائه می‌دهد که شامل، زمینه جدیدی در تحقیق برای به کارگیری عملگرهای ژنتیکی برای موازی‌سازی می‌شود. متعادل‌سازی بار و حفظ توان دو هدف اساسی بودند که باید مورد توجه قرار گیرند و همچنین یافته‌ها هزینه زمان پردازش را کاهش می‌دهند، سیستم پیشنهادی از نظر هر دو هدف از سیستم پیشنهادی برتری دارد. علاوه بر این، ظرفیت تطبیقی الگوریتم پیشنهادی در پیکربندی‌های ساختاری زیرلایه مختلف ارزیابی شد. هدف متعادل سازی بار به دست آمد. استفاده از مرکز داده معمولی بالاتر از روش های دیگر بود و به مقدار بهبود یافته ای رسید.

۷- منابع

1. Khan, S., et al., Load balancing in grid computing: Taxonomy, trends and opportunities. Journal of Network and Computer Applications, 2017. 88: p. 99-111.
2. Kollati, V.K., Scheduling in Grid Computing. https://www.researchgate.net/profile/Vijaya-Kollati?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uRGV0YWlsIiwicGFnZSI6InB1YmxpY2F0aW9uRGV0YWlsIn19, 2020.
3. Sathish, K., Resource allocation and scheduling in Grid computing: Novel strategies for Resource allocation and scheduling. Publisher : LAP LAMBERT Academic Publishing (Feb. 23 2022) 2022.
4. Esmaili, S. and K. Kianfar, Grid Scheduling Considering Energy Consumption Management and Quality of Service. Journal of Grid Computing, 2022. 20(3): p. 30.
5. Singh, M. and J.D. Kumar, WBFAS: Workflow based Failure-Aware Scheduling in Grid Computing. International Journal of Intelligent Systems and Applications in Engineering, 2023. 11(1): p. 208-214.
6. Pujiyanta, A. and F. Novianto, Job Scheduling on Grid Computing Using First Fit, Best Fit, and Worst Fit. Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika, 2022. 8(2).
7. Ardi Pujiyanta, L.E.N., - Widyawan, Job Scheduling Strategies in Grid Computing. International Journal on Advanced Science, Engineering and Information Technology, Vol. 12 (2022) No. 3, pages: 1293-1300, DOI:10.18517/ijaseit.12.3.10147, 2022.
8. Eng, K., et al., An Estimation-Based Dynamic Load Balancing Algorithm for Efficient Load Distribution and Balancing in Heterogeneous Grid Computing Environment. Journal of Grid Computing, 2023. 21(1): p. 7.
9. Wided, A. and K. Okba, A new agent based load balancing model for improving the grid performance. Multiagent and Grid Systems, 2020. 16(2): p. 153-170.
10. ADNANE, A., et al., Grid Self-Load-Balancing: the Agent Process Paradigm. DOI: https://doi.org/10.15866/irecos.v12i2.12718 2017.
11. Wided, A., A New Load Balancing Algorithm with Fuzzy Logic Controller in Grid Computing. International Journal of Performability Engineering, 2022. 18(12).
12. Kaviarasan, R., G. Balamurugan, and R. Kalaiyarasan, Effective load balancing approach in cloud computing using Inspired Lion Optimization Algorithm. e-Prime-Advances in Electrical Engineering, Electronics and Energy, 2023. 6: p. 100326.
13. Yazdani, R., et al., Minimizing total absolute deviation of job completion times on a single machine with maintenance activities using a Lion Optimization Algorithm. Sustainable Operations and Computers, 2022. 3: p. 10-16.
14. Adaikalaraj, J.R. and C. Chandrasekar, To improve the performance on disk load balancing in a cloud environment using improved Lion optimization with min-max algorithm. Measurement: Sensors, 2023: p. 100834.

معصومه اکبری امامی:
کارشناسی ارشد مهندسی نرم افزار گرایش شبکه‌های کامپیوتری دانشگاه آزاد اسلامی واحد زنجان و نشانه رایانامه ایشان:
m.akbariiraninsurance@gmail.

روش ارجاع: م. اکبری امامی. ارائه یک مدل زمانبندی کارها در شبکه‌های گرید مبتنی بر الگوریتم بهینه سازی شیر. دو فصلنامه محاسبات و سامانه های توزیع شده، سال هشتم، شماره۲، شماره پیاپی۱۶، صفحه ۱۱۴ تا ۱۳۰، سال ۱۴۰۴.

How to cite: M.Akbari emami, Presenting a task scheduling model in grid networks based on the Lion Optimization Algorithm, Journal of Distributed Computing and Systems (JDCS), Vol 8, Issue 2, Page 114-130, 2026.

Presenting a task scheduling model in grid networks based Lion on the Optimization Algorithm

M.Akbari emami

Islamic Azad University, Zanjan Branch

Abstract

The load balancing challenge in cloud computing is very important for the effective management of cloud resources and it requires distributing the incoming network traffic or computational workload among multiple servers in a way that no server is overloaded, hence improving resource utilization, increasing throughput and reducing the response time of load balancing is crucial for distributed systems to achieve high availability and fault tolerance. In this paper, a task scheduling model in grid networks based on the Lion on the Optimization is presented. The aim of this paper is to present an improved Lion on the Optimization metaheuristic approach to solve load balancing problems in the cloud. In fact, the proposed method addresses the challenge of a single-machine scheduling problem in which the machine must be maintained after a fixed periodic interval. The aim is to minimize the total absolute deviation of the completion times for the problem under consideration. Hence, to solve this problem, a new metaheuristic algorithm, namely Lion on the Optimization, was used in the proposed method, and the comparative results showed the absolute superiority of our proposed algorithm over the base paper.

Key words: Job scheduling, grid networks and the Lion Optimization Algorithm.