

بهبود زمان بندی وظایف ایستا در سیستم های توزیع شده نامتقارن با استفاده از الگوریتم بهینه سازی مبتنی بر آموزش یادگیری

افشین شهرکی مقدم

استادیار گروه کامپیوتر، دانشگاه غیر انتفاعی هاتف، سیستان و بلوچستان، زاهدان

نیما ریگی کارواندار

دانشجوی کارشناسی ارشد رشته مهندسی فناوری اطلاعات، دانشگاه غیر انتفاعی هاتف، سیستان و بلوچستان، زاهدان

چکیده

به منظور بهبود کارایی سیستم های توزیع شده، ارائه الگوریتم های زمان بندی از اهمیت بالایی برخوردار است. زمان بندی وظایف ایستا در محیط های توزیع شده ناهمگن یک مسئله بهینه سازی NP-Hard است. استفاده از الگوریتم های فرامکاشفه ای برای حل این مسائل استفاده می شود. در این تحقیق، یک الگوریتم زمان بندی وظایف ایستا در محیط ناهمگن توزیع شده جدید مبتنی بر الگوریتم بهینه سازی آموزش و یادگیری ارائه پیشنهاد شده است. در الگوریتم پیشنهاد شده با در نظر گرفتن اولویت و زمان اجرای هر وظیفه و تخصیص نزدیکترین پردازنده به وظیفه مورد نظر ارائه شده که زمان اجرای کل وظایف حداقل می شود. همگرایی روش پیشنهاد شده با تعداد تکرار کمتر برای رسیدن به جواب نسبتا بهینه به نسبت دیگر الگوریتم های فرامکاشفه ای انجام می شود. به منظور ارزیابی، مقایسه ای جامع در پارامترهای ارزیابی شامل توافق سطح سرویس، میزان مصرف انرژی، مقدار حافظه استفاده است. نتایج ارزیابی نشان می دهد که در روش پیشنهادی در مقایسه با روش های مشابه، مقدار مصرف انرژی با خاموش کردن میزبان ها و ماشین های فیزیکی به حداقل مقدار خود در مقایسه با سایر روش ها رسیده است. این در حالی است که توافق سطح سرویس و نیز حافظه مصرفی هم در حالات متعدد ارزیابی بهبود پیدا کرده است.

واژگان کلیدی: محاسبات ابری، زمان بندی وظایف، الگوریتم بهینه سازی مبتنی بر آموزش یادگیری، مهاجرت ماشین مجازی

۱-مقدمه

محاسبات ابری^۱، تکنولوژی نسبتاً جدیدی در چرخه فناوری اطلاعات می باشد و به عنوان نسل جدید، محاسبات ابری الگویی است که قابلیت های مرتبط با فناوری اطلاعات را به عنوان سرویس به کاربران خود عرضه می کند [۱، ۲]. این تکنولوژی محاسبات ابر، مزایای زیادی را برای کاربران خود فراهم می کند، اما با چالش هایی نیز روبرو است. یکی از این چالش ها الویت بندی در زمان بندی^۲ موثر وظایف در مراکز داده ابر است [۳].

مجازی سازی^۳ تکنولوژی نرم افزاری است که موجب دگرگونی در چشم انداز فناوری اطلاعات و همچنین تغییرات اساسی در زیرساخت IT سازمان را فراهم می سازد. مجازی سازی به شما اجازه اجرای چند نرم افزار کاربردی و سیستم عامل را بر روی یک کامپیوتر می دهد [۴]. مجازی سازی به افزایش بهره وری سخت افزار، کاهش هزینه، کاهش مصرف انرژی و سازمادهای هزینه ها کمک می کند که باعث حفظ سرمایه سازمان می شود. مجازی سازی یک تکنیک طراحی اساسی در تمامی معماری های ابر است. مجازی سازی اجازه می دهد که سرورها، وسایل ذخیره سازی و دیگر سخت افزارها به صورت یک مخزن از منابع در مقایسه با سیستم های جدا از هم رفتار کنند، بنابراین این منابع می توانند به مجرد تقاضا تخصیص داده شوند [۵].

مسئله الویت بندی یا زمان بندی مساله مهم با درجه ی سختی NP در زمینه مدیریت منابع در سیستم های محاسبات ابر می باشد، زیرا تعداد وظایف هر ماشین و مدت زمان اجرای آنها در محیط ابر به سرعت تغییر می کنند [۶]. محاسبه تمام نگاشت های ممکن بین تعداد ماشین های مجازی و مکان های موجود برای ماشین های فیزیکی (PM)^۴ و هاست ها در محیط ابری بسیار دشوار است و یافتن یک نگاشت بهینه اصلاً کار ساده ای نیست و نیاز به الگوریتم های بهینه سازی دارد که بتواند توزیع ماشین های مجازی روی ماشین های فیزیکی PM را به صورت کارآمد انجام دهد تا اهداف کیفیت سرویس کاربر و سیستم که همان زمان پاسخ مناسب با کمترین میزان انرژی مصرفی و همچنین کمترین حافظه مورد نیاز است، کاهش یابد [۷]. یکی از راه های الویت بندی مهاجرت ماشین مجازی و سپس مکان یابی آنها بر اساس یک الگوریتم بهینه است. با مهاجرت ماشین مجازی، تعداد کمتری از ماشین های فیزیکی در شرایط بار کاری بسیار زیاد یا بار کاری بسیار کم قرار بگیرند [۸]. پس از مهاجرت ماشین و ایجاد نگاشت بهینه بین ماشین های مجازی و فیزیکی، انجام عملیات متعادل سازی^۵ بار و کاهش مصرف انرژی با خاموش کردن ماشین هایی که دیگر در آنها باری وجود ندارد، صورت خواهد گرفت. از آنجایی که مصرف انرژی در یک ماشین فیزیکی در حالت بیکاری در حدود ۶۰ درصد حالت کاری تمام باری آن است، با خاموش کردن ماشین های کم کار و انتقال به سایر ماشین ها این بهینه سازی صورت خواهد گرفت [۹-۱۱]. در این شرایط وظایف موجود بر روی ماشین های مجازی با شرایط بار کاری بسیار زیاد، در حالت خود باقی می ماند و ماشین های مجازی به ماشین های فیزیکی دارای بار متعادل و یا کم بار انتقال داده خواهد، و انرژی مصرفی کنترل خواهد شد. زیرا اگر انرژی که در مراکز داده استفاده می شود کنترل نگردد، آنگاه در گام اول روی کیفیت سرویس^۶ کاربران تأثیر خواهد گذاشت [۱۲].

زمان بندی ماشین های مجازی بر روی ماشین های فیزیکی در سال های اخیر یک چالش تحقیقاتی عمده بوده است و نتایج نگاشت منابع مجازی بر روی منابع فیزیکی تأثیر مهمی بر استفاده بهینه و کارآمد از منابع و متعاقب آن کاهش هزینه های عملیاتی یک مرکز داده دارد [۱۳]. در خصوص این مسئله تاکنون روش های بسیاری پیشنهاد شده است که در برخی از آنها روش های ریاضی و احتمالاتی استفاده می شود و در برخی دیگر که اخیراً رواج یافته اند روش های

¹ Cloud Computing

² Task Scheduling

³ Virtualization

⁴ Physical machine

⁵ Balancing

⁶ Quality Of Service

مکاشفه ای^۷ یا فرا مکاشفه ای^۸ استفاده می شود [۱۴]. در تمامی روش های پیشنهادی تلاش بر این است که متعادل سازی بر روی ماشین های فیزیکی بدست آید. متعادل نمودن بار به اجرای وظیفه در حداقل زمان ممکن کمک می کند و عملکرد سیستم را افزایش می دهد، یعنی کاربر می تواند پاسخ خود را در کوتاهترین بازه زمانی ممکن دریافت کند. بطوریکه اهداف کاهش زمان پاسخ سیستم که یکی از اهداف کیفیت سرویس است. افزایش قابلیت اطمینان سیستم و یا ترکیب آنها برآورده شود [۱۵].

همانطور که بیان شد در مسئله زمان بندی وظایف ایستا در سیستم های توزیع شده نامتقارن تخصیص اولویت به وظایف هم برای الگوریتم های هیوریستیک و هم برای الگوریتم های جستجو حائز اهمیت زیادی است [۱۶، ۱۷]. تخصیص اولویت به وظایف همچنین تاثیر بسزایی در نتایج زمان بندی و زمان پردازش موازی واقعی دارد. در روش های اخیر، در تخصیص اولویت به وظایف تنها زمان پردازش هر وظیفه در نظر گرفته می شود. دلیل این امر این است که در نظر گرفتن زمان ارتباطات بین وظایف، خود منجر به افزایش دامنه مورد جستجو برای یافتن راه حل های کاندید می شود. اگرچه در موارد عملی اجرای موازی وظایف، چنانچه دو وظیفه بر روی پردازنده های متفاوتی اجرا شوند، سربار ناشی از ارتباطات آن ها باید در نظر گرفته شود. بنابراین انتظار می رود تا نتایج بهتر زمان بندی در همان مراحل اولیه جستجو بدست آید [۱۸].

با توجه به نقش سیستم های توزیع شده در محاسبات با کارایی بالا. در این تحقیق، یک روش زمان بندی جدید با استفاده از الگوریتم بهینه سازی مبتنی بر آموزش و یادگیری ارائه می گردد که یکی از مهم ترین ویژگی های این الگوریتم، عدم وابستگی به پارامترهاست. چرا که این الگوریتم، کمترین تعداد پارامتر ممکن را دارد و از این نظر، می تواند یک امتیاز ویژه را داشته باشد. برای اینگونه سیستم ها که دارای عدم یکپارچگی هستند میتواند مورد توجه قرار بگیرد به اینصورت که با در نظر گرفتن اولویت و زمان اجرای هر وظیفه و تخصیص نزدیکترین پردازنده به وظیفه مورد نظر ارائه شده که زمان اجرای کل وظایف را به حداقل برساند و همچنین تعداد تکرارها برای رسیدن به جواب نسبتا بهینه به نسبت دیگر الگوریتم های فرامکاشفه ای را کاهش می دهد. در ادامه این مقاله پس از تشریح پیشینه تحقیق، روش پیشنهادی ارائه خواهد شد. سپس نتایج شبیه سازی روش پیشنهادی آورده خواهد شد. در نهایت نتیجه گیری مقاله بیان می گردد.

۲- پیشینه تحقیق

سلوارانی^۹ و همکاران الگوریتم مبتنی بر هزینه برای نگاشت کارآمد وظایف به منابع در دسترس سیستم توزیع شده ارائه کرده اند. این الگوریتم زمان بندی هم هزینه استفاده از منبع و هم کارایی سیستم را اندازه گیری می کند [۱۹]. مزماز^{۱۰} و همکاران یک الگوریتم زمان بندی ژنتیک ترکیبی دو هدفه برای برنامه های کاربردی موازی اولویت محدود در سیستم های توزیع شده ناهمگن مانند زیرساخت سیستم های توزیع شده، پیشنهاد داده اند [۲۰]. مانتوس^{۱۱} و همکاران در مقاله ای یک الگوریتم زمان بند جدید مبتنی بر الگوریتم کلونی مورچه را ارائه نموده اند. هدف این زمان بند کمینه کردن زمان گردش مجموعه کارها هم زمان با کمینه کردن حداکثر زمان اتمام انجام کل کارها می باشد [۲۱]. پوللا^{۱۲} و همکاران الگوریتم زمان بندی کار را بر اساس محدودیت های زمان و

⁷ Heuristic

⁸ Metaheuristic

⁹ Selvarani

¹⁰ Mezmaz

¹¹ Allahverdi

¹² Poola

هزینه برای محاسبات ابری پیشنهاد کرده‌اند، که می‌تواند زمان اتمام کل و هزینه تحت محدودیت زمان را به حداقل برساند [۲۲]. الله وردی^{۱۳} و همکاران یک روش برنامه ریزی کار استاتیک بر اساس الگوریتم بهینه‌سازی ذرات (PSO¹⁴) ارائه داده است، که در آن وظایف غیر پیشگیرانه و مستقل فرض شده است [۲۳] صالحی^{۱۵} و همکاران الگوریتم MOC را برای حل مسئله زمان‌بندی پویا با محدودیت زمان در محاسبات توزیع ناهمگن، پیشنهاد کردند [۲۴]. کوماری^{۱۶} و همکاران با ارائه یک طبقه بندی که شامل تحولات اخیر بویژه موارد موجود در محاسبات ابری می‌باشد، طبقه بندی مسئله زمان‌بندی در سیستم‌های توزیع شده را معرفی می‌کند [۲۵]. عبادی فرد^{۱۷} و همکاران برای غلبه بر کاستی‌های برنامه زمان‌بندی وظیفه موجود و رویکردهای متعادل سازی بار، یک رویکرد جدید را پیشنهاد می‌کند که از خوشه‌بندی توالی غالب (DSC¹⁸) برای برنامه ریزی کار و الگوریتم اتصال حداقل وزن (WLC¹⁹) برای تعادل بار استفاده می‌کند [۲۶]. دمیرک^{۲۰} و همکاران مسئولیت برنامه ریزی یک محاسبات موازی را با یک سیستم موازی توزیع شده با محدودیت منابع کلی به طور خاص یک بودجه کلی انرژی و منابع قابل اطمینان را بررسی کرده است، و این امکان را برای طیف وسیعی از مبادلات قدرت / عملکرد مختلف فراهم می‌کند. [۲۷]. لی^{۲۱} و همکاران یک روش زمان‌بندی منابع برای محاسبات مه را پیشنهاد می‌کند. ابتدا ویژگی‌های منابع را استاندارد و عادی می‌کنند، سپس روش‌های خوشه‌بندی فازی را با بهینه‌سازی ذرات برای تقسیم منابع ترکیب می‌کنند و در نهایت مقیاس جستجوی منابع کاهش می‌یابد [28]. زیات^{۲۲} و همکاران یک تحقیق عمیق از الگوریتم‌های مختلف فرا ابتکاری و فرامکاشف‌هایی TS ارائه کرده‌اند. الگوریتم بهینه‌سازی ذرات (PSO²³) و الگوریتم‌های ژنتیکی (GA²⁴) به عنوان الگوریتم‌های فرا ابتکاری پیاده سازی شده و عملکرد آنها با تکنیک‌های فرامکاشف‌هایی (FCFS²⁵) و (SJF²⁶) در محیط متقارن و نامتقارن مقایسه شده است [۲۹].

۱-۱- روش پیشنهادی

هدف اصلی این پژوهش ارائه روشی برای زمان‌بندی کارها در سیستم‌های توزیع شده ناهمگن (محاسبات ابری) است. برای این کار از الگوریتم بهینه‌سازی مبتنی بر آموزش یادگیری بهبود یافته استفاده خواهد شد. در این قسمت روند کلی روش پیشنهادی ارائه می‌گردد. در این پژوهش از الگوریتم مذکور در دو مرحله استفاده می‌شود. در ادامه مراحل روش پیشنهادی به صورت دقیق بیان خواهد شد. این روند در شکل ۱ هم نشان داده شده است. روال روش پیشنهادی به صورت زیر است:

- شروع
- دریافت اطلاعات از سیستم‌ها و فرایندها

¹³ Allahverdi

¹⁴ particle swarm optimization

¹⁵ Salehi

¹⁶ kumrai

¹⁷ Ebadifard

¹⁸ dominant sequence clustering

¹⁹ weighted least connection

²⁰ Demirci

²¹ Li

²² Ziyath

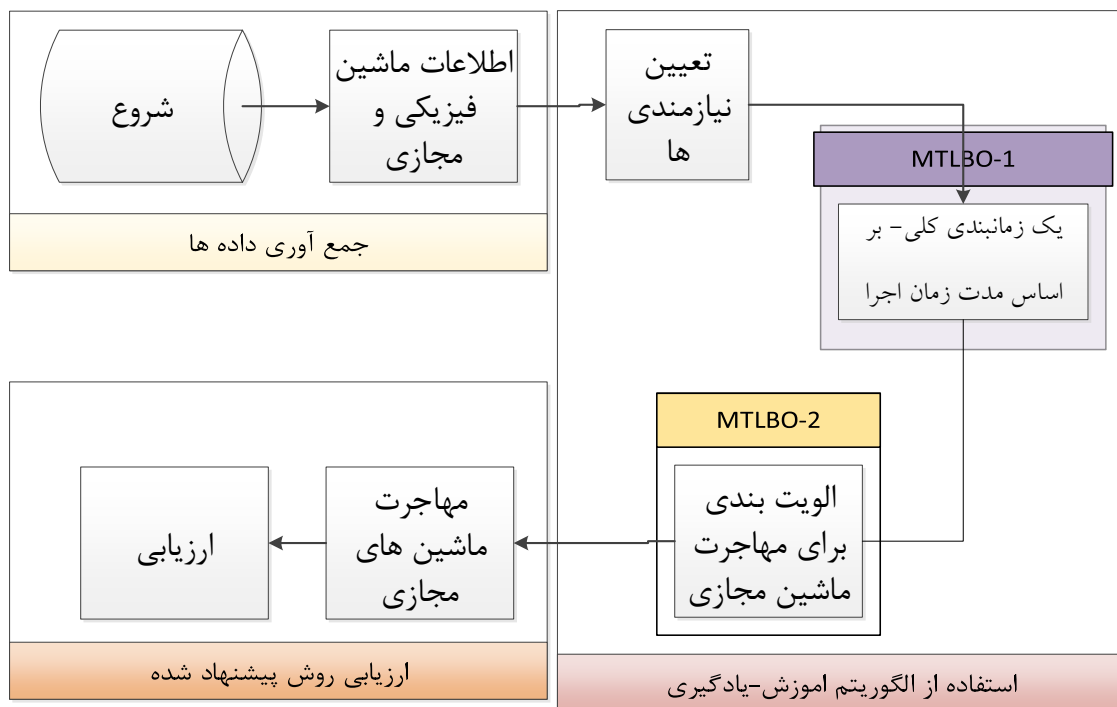
²³ Particle Swarm Optimization

²⁴ Genetic Algorithms

²⁵ First Come First Serve

²⁶ Shortest Job First

- تعیین نیازمندی هر فرآیند
- گام اول استفاده از MTLBO: در این گام تعیین می شود که کدام دسته کارها به کدام سیستمها اختصاص داده شوند (یک زمانبندی کلی - بر اساس مدت زمان اجرا)
- گام دوم استفاده از MTLBO: در این مرحله که روی هر سیستم (ابر) اجرا می شود، زمانبندی کارها صورت می گیرد.
- محاسبه معیارهای ارزیابی
- پایان



شکل (۱) بلوک دیاگرام روش پیشنهادی

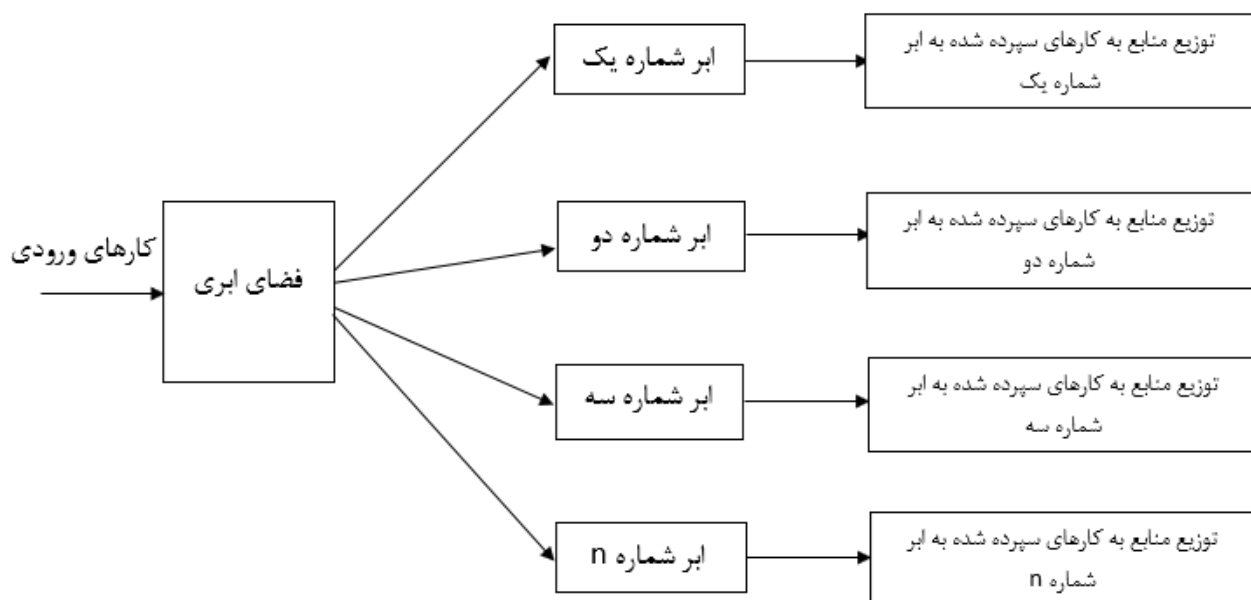
بر اساس آنچه که در ابتدای این بخش آورده شده است، روند پیشنهادی پژوهش مبتنی بر زمانبندی دو مرحله ای به این صورت است که، فرآیندها یا وظایف ورودی را دو به دو به بخش اصلی تقسیم کرده تا کارایی نهایی افزایش پیدا کند. روش پیشنهادی به صورت متوالی اجرا می شود. لذا به طور کلی:

- در استفاده برای بار اول از MTLBO کارها بین ماشینها (سیستم-ماشین مجازی) توزیع می شود.
- در استفاده دوم، کارهایی که برای هر ماشین مجازی در نظر گرفته شده است، زمانبندی می شوند و ترتیب استفاده از آنها معین می شود.

روند کلی بیان شده به صورت شکل ۲ است.

توزیع کارها روی ابر در
مرحله اول با MTLBO

توزیع کارها روی منابع در مرحله
دوم با MTLBO



شکل (۲) روند کلی زمانبندی کارها به کمک در روش پیشنهادی

در ادامه جزئیات روش پیشنهادی شرح داده خواهد شد.

۳-۱- توزیع کارها بین منابع ابری موجود در گام اول

برای شروع متعادل کننده بار اطلاعات را برای اختصاص بار کاری جمع آوری می کند. با جمع آوری اطلاعات اولیه در واقع نیاز سنجی به زمان بندی انجام می شود. که براساس اطلاعات جمع آوری شده، گام دوم، تصمیم گیری برای محاسبه بهترین توزیع داده ممکن صورت می گیرد. در نهایت در آخرین مرحله داده ها از ماشین فیزیکی، یا هاستی که روی آن اضافه بار وجود دارد به سمت هاست هایی که روی آن بار زیادی وجود ندارد انتقال داده می شود. حال اگر ویژگی های هاست مورد نظر بهینه نشود، انتقال به سایر هاست های پردازنده داده می شود. بنابراین برای کاهش احتمال اختصاص ناخواسته، مفاهیم بهینه سازی با زمان بندی در محاسبات ابر استفاده شده است.

در این بخش روند توزیع کارها مابین ابرها بیان می شود. در این حالت فرض می شود تعداد ابرها ثابت است. و قرار است کارها بین ابرها تقسیم شوند. لذا در این حالت موارد زیر اهمیت دارد.

الف- نحوه نمایش جواب های پیشنهادی MTLBO

ب- نحوه محاسبه معیارهای کارهایی جواب ها

در ادامه هر کدام از این موارد توضیح داده می شود.

۳-۲- نمایش عامل‌ها

نمایش جواب‌ها برای حل مساله در الگوریتم مذکور یا MTLBO به تعداد کارها و تعداد ابرها بستگی دارد. بر همین اساس اگر m بیانگر تعداد کارها و n بیانگر تعداد ابرهای موجود در سیستم باشد، تعداد m کار باید به نحوی در بین n ابر توزیع شوند تا کارایی زمان‌بندی بهبود یابد. لذا شکل زیر، نمایش جواب‌ها در گام اول را نشان می‌دهد.

شماره کار	۱	۲	۳	۴	...	i	...	$m-1$	m
شماره ابر	۱/۱	۵/۴	۳/۲	۲/۵	...	$J+0.3$...	$N+0.4$	۳/۵

شکل (۳) نمایش عامل‌ها در فاز اول

با توجه به اینکه شماره ابر باید مقداری گسسته باشد، باید شکل ۳ تغییر کند. باید دقت کرد که بر اساس شکل ۴ شماره‌ی ابر به صورت ضمنی در هر جواب قرار دارد. بنابراین هر جواب یک بردار به طول m است. بر اساس موارد گفته شده شکل ۴ توزیع کارها بین ابرها در شکل ۴ نشان داده شده است:

شماره کار	۱	۲	۳	۴	...	i	..	$m-1$	m
شماره ابر	۱	۵	۳	۲		j		N	۳

شکل (۴): نحوه‌ی استفاده از عامل‌ها جهت نمایش توزیع کارها بین ابرها

شکل ۴ با گرفتن جز صحیح از شکل ۴۴ حاصل می‌گردد. بر اساس شکل ۴:

- کار شماره ۱ در ابر شماره ۱ انجام می‌شود.
 - کار شماره ۲ در ابر شماره ۵ انجام می‌شود.
 - کار شماره ۳ در ابر شماره ۳ انجام می‌شود.
 - به همین ترتیب کار شماره i در ابر شماره j و کار شماره $m-1$ در ابر شماره n انجام می‌شود.
- پس از اینکه این توزیع صورت گرفت، باید کارهایی هر عامل محاسبه گردد.

۳-۳- تعریف تابع هدف در گام اول

برای محاسبه مقدار تابع هدف برای هر جواب ابتدا باید به کمک فاز دوم، توزیع کارها در هر ابر صورت بگیرد و بهترین توزیع برای حالت دوم بیانگر کارایی در حالت اول خواهد بود.

$$(۱) \quad Perf(Step1) = \sum_{i=1}^n BestAnsStep2(Step1 \text{ answer})$$

در ادامه نحوه‌ی استفاده از الگوریتم MTLBO توضیح داده می‌شود.

۳-۴- زمانبندی کارها در ابر در گام دوم

در این بخش نحوه توزیع کارها در منابع موجود در ابر توضیح داده می شود. در این بخش دو نکته باید در نظر گرفته شود:

- هر کار توسط کدام منبع در ابر باید اجرا شود (تخصیص کارها به منابع موجود در ابر)
 - ترتیب انجام کارها در هر منبع به چه صورت خواهد بود (زمانبندی کارها در منابع هر ابر)
- همانند همه الگوریتم های فراابتکاری برای حل مسائل برای این مساله نیز باید نحوه نمایش عامل ها و نیز نحوه محاسبه برازندگی باید معین گردد. در ادامه در این باره بحث خواهد شد.

۳-۵- نمایش جواب ها در فاز دوم

در این قسمت نحوه نمایش جواب ها برای توزیع کارها در ابر بیان می شود. در این مرحله فرض می شود که برای ابر شماره i که دارای k منبع داده است، h کار اختصاص می یابد. بنابراین توزیع باید به نحوی در منابع ابری صورت گیرد که معیارهای کارایی بهبود یابند. نمایش جواب ها در این مرحله شبیه به مرحله قبل است، اما با اندکی تغییر در آن. چرا که نمایش جواب ها و برداشت از آنها باید به نحوی باشد که ترتیب اجرایی که به هر منبع اختصاص داده می شود، نیز تعیین شود. برای نمایش جواب ها در این مرحله مانند شکل ۵ عمل می شود. در این شکل هر هر جواب دارای طول h (تعداد کارهای اختصاص داده شده با آن ابر) است و مقادیر بیانگر کاری است که باید توسط هر کدام از منابع داده اجرا گردد.

شماره کار	۱	۲	۳	۴	...	i	..	$m-1$	m
شماره ابر	۳/۶	۱/۲	۱/۹	۳/۳		۲/۳		۲	۱/۱

شکل (۵) نمایش جواب در گام دوم

در محیط زمان بندی منابع هر ابر به صورت زیر خواهد بود:

شماره کار	۱	۲	۳	۴	...	i	..	$m-1$	m
شماره ابر	۳	۱	۱	۳		۲		۲	۱

شکل (۶) نحوه نمایش جواب های برای زمان بندی کارها در هر ابر

شکل ۶ با گرفتن جز صحیح (روند کردن به سمت پایین) از شکل ۵ حاصل می گردد. بر اساس شکل ۶ خواهیم داشت:

- کار شماره ۱ در منبع داده ۳ انجام می شود.
- کار شماره ۲ در منبع داده ۱ انجام می شود.
- کار شماره ۳ در منبع داده ۱ انجام می شود.

اما نکته‌ای که در این قسمت اهمیت دارد این است که چگونه ترتیب انجام کارها یا زمانبندی کارها در هر ابر بر اساس نمایش بالا تعیین گردد. بر همین اساس با در نظر گرفتن شکل ۵ می‌توان گفت کارهای شماره ۲، ۳ و h باید در منبع شماره ۱ موجود در ابر اجرا شوند، حال بر اساس شکل ۵ داریم:

مقدار عامل در موقعیت ۲ برابر ۱.۲، در موقعیت ۳ برابر ۱.۹ و در موقعیت h برابر ۱.۱ است. برای اینکه مشخص شود کدام کار باید زودتر انجام شود، این مقادیر را به صورت صعودی مرتب کرده و بر اساس ترتیب به وجود آمده هر کار که دارای مقدار کمتری باشد، زودتر اجرا خواهد شد، لذا برای مثال مذکور خواهیم داشت:

ترتیب به صورت صعودی و نیز ترتیب اجرا:

- کار شماره h با مقدار ۱.۱ (اجرا به عنوان اولین کار در منبع مذکور)
- کار شماره ۲ با مقدار ۱.۲ (اجرا به عنوان دومین کار)
- کار شماره ۳ با مقدار ۱.۹ (اجرا به عنوان سومین کار در منبع شماره یک)

بدین ترتیب این مشکل نیز حل خواهد شد. پس از اینکه این توزیع صورت گرفت، باید کارهایی هر عامل محاسبه گردد.

۴-ارزیابی روش پیشنهادی

در این مقاله مسئله زمان بندی زمان بندی وظایف ایستا در سیستم‌های توزیع شده ناهمگن در محاسبات ابر جهت کاهش مصرف انرژی مبتنی بر الگوریتم MTLBO ارائه شده است. در فصل چهارم روش پیشنهادی به صورت کامل تشریح شد. در این بخش روش پیشنهادی با معیارهای ارزیابی متداول، مورد بررسی قرار خواهد گرفت.

۴-۱-پارامترهای شبیه سازی

برای ارزیابی روش پیشنهادی، این روش در نرم افزار متلب شبیه سازی شده است. در ابتدا پارامترهای مورد نیاز برای ارزیابی مورد بررسی قرار داده می‌شود. برای ارزیابی روش پیشنهادی بر اساس شبیه ساز متلب محاسبه و بررسی نقض کیفیت سرویس و مقایسه مصرف انرژی روش پیشنهادی و روش الگوریتم مورچگان و همچنین زنبور عسل پرداخته می‌شود. برای ارزیابی روش پیشنهاد شده در زمان بندی و همچنین بهینه سازی VM ها شرایط زیر برای PM ها و VM ها بر اساس [۳۰] در نظر گرفته شده است.

هشت حالت در نظر گرفته شده است که در هر حالت تعداد ماشین‌های مجازی از ۱۰۰ تا ۲۰۰۰ ماشین متغیر است و با A_1 تا A_8 نشان داده می‌شوند. هر PM دارای یک CPU دارای ۱۶ هسته و ۴ گیگا RAM است. هر VM به ۱ تا ۴ هسته و همچنین ۰.۵ تا ۲ گیگا فضای RAM جهت نشستن نیاز دارد. این شرایط به صورت تصادفی ایجاد می‌شود. جدول احتمالات ماشین‌های مجازی به صورت جدول (۱) است. همچنین نسبت تعداد ماشین مجازی به ماشین‌ها فیزیکی ۱۰ به ۸ است. یعنی به ازای هر ۱۰ ماشین مجازی ۸ ماشین فیزیکی وجود دارد [۳۰].

جدول (۱) احتمالات مربوط به ماشین‌ها

تعداد هسته	احتمال
۴ هسته	۰/۲۵
۳ هسته	۰/۱۲۵
۲ هسته	۰/۱۲۵

در این حالت CPU محدودیت منبع است و نسبت CPU به RAM بایستی ۸:۱۰ باشد. راه حل بهینه به صورت تصادفی تولید می شود و البته ناشناخته است. در نتیجه حدهای پائین راه حل بهینه به عنوان بیشترین نرخ مجموع VM ها به ظرفیت CM ها در نظر گرفته خواهد شد.

در شبیه سازی روش پیشنهادی، یک ابر با تعدادی سرویس دهنده فیزیکی و یک کنترلر مرکزی وجود دارد. شرح رفتار شبکه به این صورت است که هر سرور دارای یک کنترلر است. همچنین یک کنترلر مرکزی که وظیفه دریافت اطلاعات سرورها از کنترلرهایی که روی هر هاست هست نیز وجود دارد، اطلاعات ارسال شده توسط کنترلرها در الویت بندی وظایف در ماشین های مجازی به کار گرفته خواهد شد. از وظایف کنترلر مرکزی این است که وضعیت ماشین های فیزیکی و هاست ها را دریافت کرده و با توجه به درخواست های دریافت شده از طرف کاربران و نیز وضعیت موجود در هر هاست فیزیکی، با اجرای $MTLBO$ ، اقدام به زمان بندی وظایف به صورت موثر کند.

در این تحقیق از الگوریتم $MTLBO$ برای حل زمان بندی بهبود زمان بندی وظایف ایستا در سیستم های توزیع شده ناهمگن از روی هاست های فیزیکی در محاسبات مه استفاده شده است. کنترلر مرکزی اطلاعات از تمام سرورهای فیزیکی و مجازی را دریافت خواهد کرد. برای حالت بی کار به طوریکه بار کاری پردازنده عملاً صفر است ۳ وات توان مصرفی و برای حالتی که بار کاری ماکزیمم است ۲۵۰ وات توان مصرفی برای یک پردازنده در نظر گرفته ایم. در این وضعیت اگر بارکاری پردازنده فیزیکی به ماکزیمم اندازه خود برسد در حالی که بار کاری واقعی تحمیل شده توسط ماشین های مجازی بیشتر از توان پردازشی باشد در این حالت به اصطلاح سربار^{۲۷} اتفاق افتاده است. در حالت سربار، کیفیت سرویس کاهش می یابد. که عملاً هدف از به کار گیری الگوریتم $MTLBO$ در جلوگیری از سربار است.

استراتژی های زمان بندی موثر زمان بندی وظایف ایستا در سیستم های توزیع شده ناهمگن به دنبال تخصیص VM به تعداد ماشین ها و سرورهای فیزیکی معین و البته کافی است که بر اساس برخی اهداف صورت می گیرد. یکی از مهم ترین اهداف زمان بندی کاهش توان مصرفی در نتیجه کاهش مصرف انرژی است، چرا که روی هزینه های عملیاتی و همچنین محیط زیست اثر گذار است. مدلسازی مسئله زمان بندی به صورت زیر انجام شده است.

یک محیط محاسبات ابری با n ماشین فیزیکی و m ماشین مجازی در نظر است. P نشان دهنده مجموعه PM ها و P_i نشان دهنده i امین PM است. که $P_i \in P$ و i به مجموعه $[1 \dots n]$ تعلق دارد. همچنین به صورت مشابه V نشان دهنده مجموعه VM ها است که V_j نشان دهنده j امین VM است، که $V_j \in V$ و j به مجموعه $[1 \dots m]$ تعلق دارد.

هدف اصلی کاهش مصرفی انرژی، کاهش توان مصرفی است. در کنار آن روی cpu و حافظه مصرفی هم تمرکز خواهد شد. توان محاسباتی مورد نیاز و حافظه مورد نیاز برای V_j با V_{cpuj} و V_{ramj} نشان داده خواهد شد. همچنین توان محاسباتی و حافظه مورد نیاز برای P_i با $P_{cpu i}$ و $P_{ram i}$ نشان داده خواهد شد.

فرض بر این است که هر PM دارای ظرفیت کافی برای هر VM تکی است به عبارت دیگر هاست ها دارای ظرفیت کافی برای پذیرش VM ها هستند. راه حل زمان بندی وظایف با S نشان داده می شود. که یک ماتریس صفر و یک مجاورت^{۲۸} است. درایه های این ماتریس x_{ij} است که اگر P_i در VM_j قرار داده شود $x_{ij} = 1$ در غیر این صورت $x_{ij} = 0$ است. در این صورت زمان بندی بهینه وظایف بدست می آید که تعداد PM های فعال در محاسبات ابر توزیع شده حداقل باشد. [۳۰]

²⁷ overload

⁴-zero-one adjacency Matrix

$$\sum_{i=1}^n y_i \quad (1)$$

که y_i تعداد PM های فعال در محیط ابر توزیع شده است. در صورتی که V_j روی P_i باشد [۳۰]:

$$x_{ij} = \begin{cases} 1 & \text{if } V_j \in P_i \\ 0 & \text{else} \end{cases} \quad (2)$$

در رابطه ۲ در صورتی که ماشین مجازی V_j روی ماشین فیزیکی P_i قرار داشته باشد. در اینصورت مقدار y_i بر اساس رابطه (۳) محاسبه خواهد شد [۳۰]

$$y_i = \begin{cases} 1; & \sum_{j=1}^m x_{ij} \geq 1 \quad \forall i \in P \\ 0; & \text{otherwise} \end{cases} \quad (3)$$

انرژی مصرفی در یک مرکز داده شامل انرژی مصرفی گره های محاسباتی (سرورها) و انرژی مصرفی در زمان مهاجرت ماشین های مجازی می باشد که به صورت فرمول زیر محاسبه می شود [۳۱]:

$$E_{total} = E_{computation} + E_{migration} \quad (4)$$

به منظور بررسی انرژی مصرف شده توسط میزبان های فیزیکی از آنجا که استفاده از پردازنده نقش بسیاری در افزایش انرژی مصرفی در مقایسه با منابع دیگر دارد، از این معیار به عنوان معیار مؤثر در مصرف انرژی استفاده می شود. طبق بررسی های به عمل آمده، یک میزبان فیزیکی بیکار به طور متوسط، تقریباً ۷۰ درصد نسبت به میزبانی که در حالت استفاده ۱۰۰ درصد از بهره پردازنده می باشد، انرژی مصرف می کند. به منظور محاسبه انرژی مصرف شده میزبان های فیزیکی از فرمول (۱۰) که در زیر بیان شده است، استفاده کرده ایم [۳۲]:

$$p(u) = k.P_{max} + (1-k).p_{max}.u \quad (5)$$

P_{max} حداکثر انرژی مصرف شده می باشد، زمانی که میزبان فیزیکی به طور کامل مورد استفاده قرار می گیرد. پارامتر k معیاری است که کسری از انرژی مصرف شده توسط میزبان فیزیکی بیکار می باشد. u بهره وری پردازنده است. P_{max} در آزمایشات به عمل آمده مقدار ۲۵۰ وات در نظر گرفته شده است که یک مقدار مناسب برای میزبان های فیزیکی جدید است. از آنجا که استفاده از پردازنده با گذشت زمان براساس اینکه بارکاری تغییر می کند به صورت انتگرال بیان می شود. در نتیجه، استفاده از پردازنده را به صورت تابعی از زمان بیان کرده اند. بنابراین مصرف کل انرژی توسط یک میزبان فیزیکی به عنوان انتگرال تابع مصرف انرژی روی یک دوره زمانی بیان شده است که در فرمول زیر (۶) ارائه شده است [۳۳]:

$$E_{computation} = \int_t p(u(t))dt \quad (6)$$

۲-۴- ارزیابی الویت پیشنهادی

جهت ارزیابی روش پیشنهادی از دو معیار استفاده می شود. معیار اول، معیار محاسبه ی کل توان مصرف شده، و معیار دوم بررسی نقض کیفیت خدمات است. در روش پیشنهادی با در نظر گرفت مهاجرت و البته الویت بندی

مهاجرت در ماشین‌های مجازی تا حدی در بهبود کیفیت سرویس و زمان اجرا مؤثرتر عمل شده است و این در حالی است که از افزایش مصرف انرژی جلوگیری شده است.

برای ارزیابی روش پیشنهاد شده در الویت‌بندی مهاجرت ماشین‌های مجازی هشت حالت در نظر گرفته شده است که در هر حالت تعداد ماشین‌های مجازی از ۱۰۰ تا ۲۰۰۰ ماشین متغیر است و با A_1 تا A_8 نشان داده می‌شوند. این شرایط به صورت تصادفی ایجاد می‌شود. همچنین نسبت تعداد ماشین‌های مجازی به ماشین‌های فیزیکی ۱۰ به ۸ است. یعنی به ازای هر ۱۰ ماشین مجازی ۸ ماشین فیزیکی به منظور مهاجرت وجود دارد. شرایط کامل برای تعیین مکان بهینه قرار گرفتن ماشین‌های مجازی در بخش بعدی ارائه شده است. این مقایسه به ازای افزایش تعداد ماشین مجازی از ۱۰۰ تا ۲۰۰۰ ماشین مجازی صورت گرفته است. ماشین مجازی مربوطه از هاست فیزیکی مبدأ به هاست مقصد، با تعیین الویت زمان بندی با MTLBO انتقال داده می‌شود.

۳-۴- مقداردهی پارامترهای TLBO و الگوریتم‌های مورد مقایسه

در این قسمت بر اساس روش تعریف شده، نحوه مقدار دهی به پارامترهای TLBO در هر مرحله توضیح داده می‌شود. مقداردهی به الگوریتم‌ها در مرحله اول:

- حداکثر تعداد تکرار، ۱۵۰ مرحله
- تعداد جمعیت، ۵۰ عدد

هدف کمینه‌سازی و برای الگوریتم دوم، حداکثر تعداد تکرار برابر ۵۰ و اندازه جمعیت برابر ۳۰ و هدف کمینه سازی است. پارامترهای شبیه‌سازی الگوریتم PSO و الگوریتم ABC در مکان‌یابی و زمان بندی بهینه در جدول (۲) و (۳) ارائه شده است. تعداد تکرار الگوریتم ۱۵۰ بار در نظر گرفته شده است.

جدول (۲) پارامترهای ورودی عامل کلونی زنبور عسل

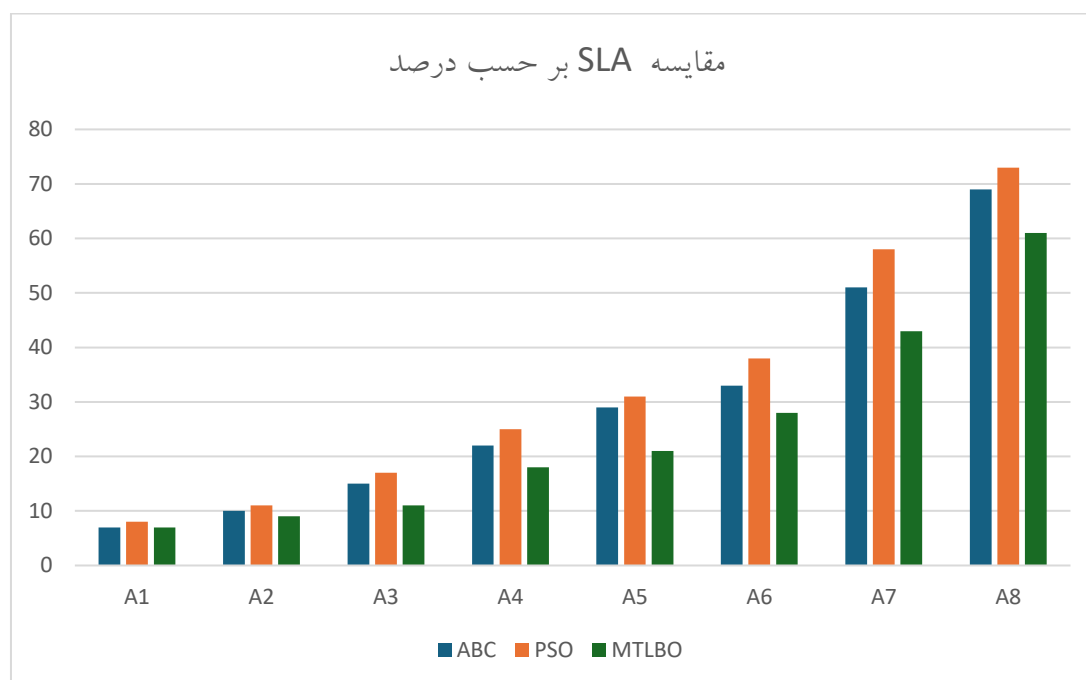
پارامتر	مقدار	توضیح
BeeNum	10	تعداد زنبورها
SourceNum	5	زنبورهای کارگر
UnlookerNum	۲	زنبورهای ناظر
MaxCycle	۲۰۰	تکرار الگوریتم

جدول (۳) تنظیم پارامترهای الگوریتم ازدحام ذرات

مقدار پارامتر	نام پارامتر
۵۰	تعداد ذره‌ها
۱۰۰	تعداد تکرار الگوریتم
۰/۵	ضریب شتاب c_1
۰/۵	ضریب شتاب c_2
۰/۷۸	ضریب اینرسی ω

۴-۴- تحلیل نتایج

با افزایش تعداد ماشین‌های مجازی مقدار SLA افزایش یافته است. این افزایش به دلیل افزایش تعداد ماشین‌های مجازی است. افزایش مقدار SLA در روش پیشنهادی MTLBO، در مقایسه با الگوریتم بهینه سازی PSO و ABC بسیار کمتر است.

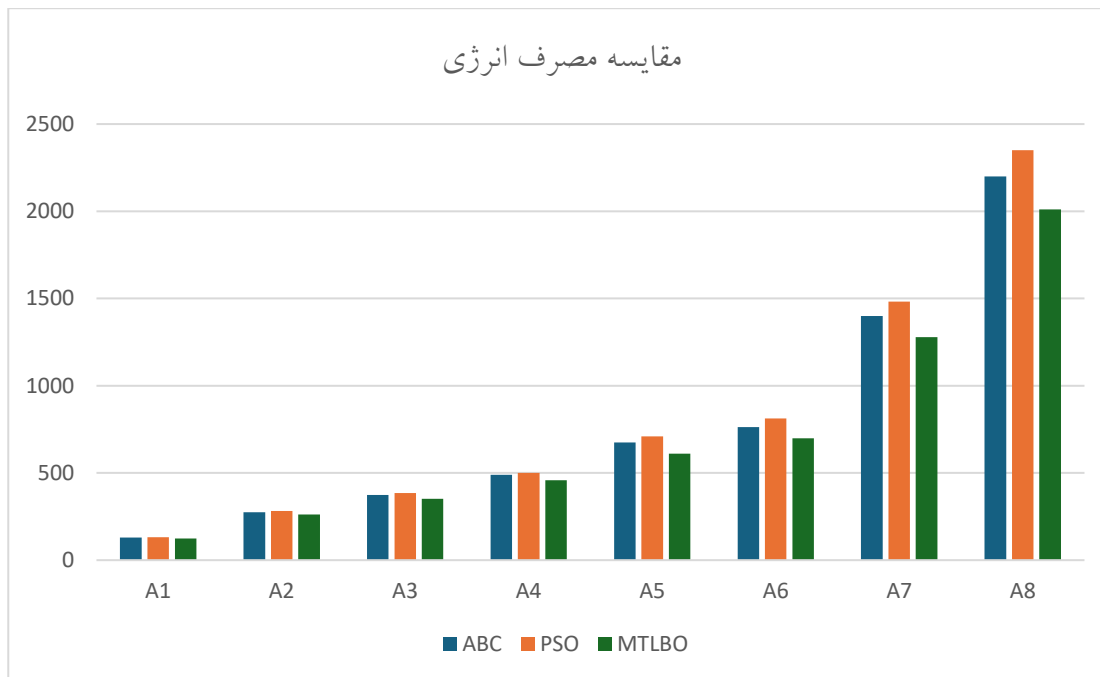


شکل (۴) مقایسه درصد نقض کیفیت سرویس SLA روش پیشنهادی MTLBO و PSO و ABC

همانگونه که در شکل (۵) نشان داده شده است، روش پیشنهادی MTLBO نسبت به روش‌های بهینه سازی ABC و PSO با افزایش تعداد ماشین‌های مجازی، نقض SLA کمتری دارد. از آنجا که در روش پیشنهادی با زمان-بندی کارآمد مهاجرت ماشین‌های مجازی صورت گرفته است، سعی کردیم از منابع میزبان‌های فیزیکی و مجازی به طور مؤثر بهره مند شویم، و همچنین با الویت‌بندی مهاجرت‌ها در ماشین‌های مجازی به کمک الگوریتم MTLBO

حداقل کردن تعداد مهاجرت‌های غیر ضروری به کمک این الگوریتم، احتمال وقوع نقض SLA نسبت به دو روش دیگر کمتر است. همچنین روش بهینه سازی ABC نتایج به مراتب بهتری نسبت به PSO دارد.

مصرف انرژی روش پیشنهادی در الویت‌بندی مبتنی بر MTLBO و روش‌های مبتنی بر PSO و ABC در شکل (۵) و همچنین بر اساس تعداد ماشین‌های مجازی متفاوت از ۱۰۰ تا ۲۰۰۰ مورد بررسی قرار گرفته است.



شکل (۵) مقایسه مصرف انرژی در الویت بندی مهاجرت ماشین مجازی

آنچه که از شکل (۸) قابل مشاهده است، با افزایش تعداد ماشین‌های مجازی، مصرف انرژی به دلیل افزایش تعداد ماشین‌های فیزیکی فعال افزایش می‌یابد. در روش پیشنهادی با زمان بندی کارآمد به کمک الگوریتم MTLBO در مهاجرت ماشین‌ها و تخصیص مؤثر با الگوریتم MTLBO سعی شده است از افزایش مصرف انرژی جلوگیری شود. مصرف انرژی به ازای تعداد هاست‌های فیزیکی درگیر، تعداد وظایف در حال اجرای روی هاست فیزیکی و تعداد ماشین مجازی بستگی دارد. این ارزیابی با الگوریتم‌های PSO و ABC نشان می‌دهد که روش ABC به مراتب بهتر از PSO عمل کرده است. هر چند با افزایش تعداد ماشین‌ها مقدار مصرف انرژی افزایش داشته است.

۵-۴- ارزیابی زمان بندی بهینه وظایف

ارزیابی روش پیشنهاد شده بهینه‌سازی زمان‌بندی وظایف‌ایستا در سیستم‌های توزیع شده ناهمگن به کمک الگوریتم MTLBO در این مرحله انجام شده است. شبیه‌سازی‌ها با روش زمان بندی مبتنی بر الگوریتم زنبور عسل و بهینه‌ساز ازدحام ذرات مقایسه شده است.

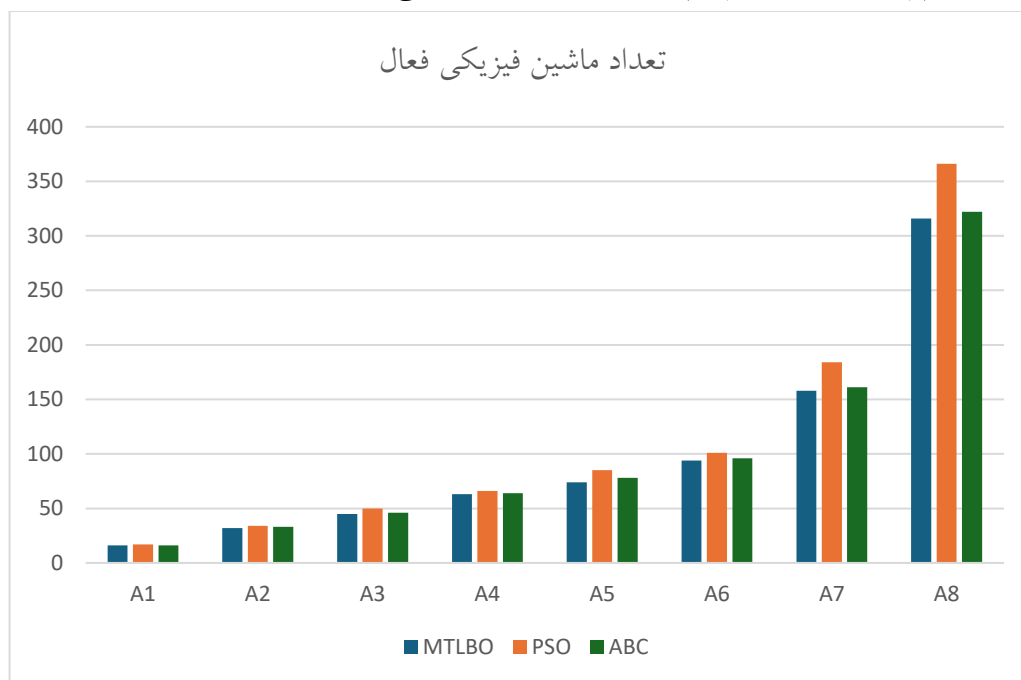
محدودیت‌های منابع در محیط ابر

جدول (۴) نتایج حاصل از شبیه‌سازی MTLBO ، PSO ، ABC را نشان می‌دهد.

جدول (۴) نتایج حاصل از محدودیت منابع در محیط ابر

حالت شبیه سازی	تعداد ماشین	MTLBO	PSO	ABC
A1	۱۰۰	۱۶	۱۷	۱۶
A2	۲۰۰	۳۲	۳۴	۳۳
A3	۳۰۰	۴۵	۵۰	۴۶
A4	۴۰۰	۶۳	۶۶	۶۴
A5	۵۰۰	۷۴	۸۵	۷۸
A6	۶۰۰	۹۴	۱۰۱	۹۶
A7	۱۰۰۰	۱۵۸	۱۸۴	۱۶۱
A8	۲۰۰۰	۳۱۶	۳۶۶	۳۲۲

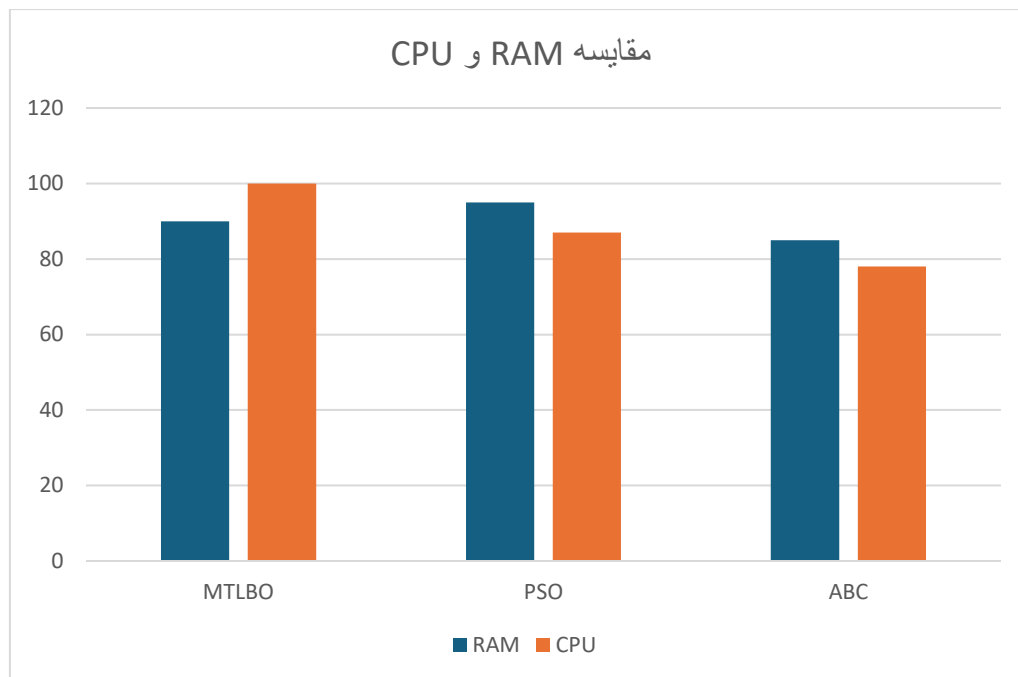
در جدول (۴) نشان دهنده PM فعال به ازای تعداد VM مد نظر است. به ازای ۱۰۰ ماشین مجازی، الگوریتم PSO ۱۷ ماشین فیزیکی و دو الگوریتم GWO و ABC ۱۶ ماشین فیزیکی فعال دارند. در حالتی که تعداد ماشین مجازی کم است، عملکرد هر سه الگوریتم تقریباً برابر است، اما زمانی که VM ها به صورت چشم گیری افزایش پیدا می کند. الگوریتم پیشنهادی به طرز چشم گیری به تعداد ماشین فیزیکی کمتری نیاز دارد.



شکل (۹) نشان دهنده تعداد PM فعال به ازای تغییرات مربوط به تعداد ماشین مجازی.

تعداد ماشین فیزیکی فعال به ازای تعداد VM در الگوریتم $MTLBO$ نسبت به PSO ، ABC کاهش یافته است.

محدودیت در مصرف CPU به اینگونه است که در حالیکه در بدترین حالت از ۹۰٪ RAM استفاده شده است و CPU به صورت کامل درگیر است، این درحالی است که در ABC با ۹۵ درصد درگیری CPU ۸۷٪ از RAM مشغول شده است و در PSO با ۸۵٪ درگیری CPU کمتر از ۷۰٪ RAM استفاده شده است. برای ارزیابی شکل (۱۰) این مقایسه را به بهترین وجه نشان می دهد.



شکل (۷) نمودار مقایسه کارکرد CPU و حافظه

همانگونه که در شکل (۷) پیدا است باز هم به ازای درگیری CPU به صورت ۱۰۰٪ مقدار کمتری از RAM در MTLBO مشغول است. همچنین از ظرفیت بهینه ماشین‌های فیزیکی بیشترین بهره‌برداری شده است. شکل ۱۰ مقایسه در حالت $M = 100$ ماشین را نشان می‌دهد.

۵- نتیجه گیری

در این تحقیق یک روش زمان‌بندی وظایف ایستا در سیستم‌های توزیع‌شده ناهمگن ارائه مبتنی بر الگوریتم بهینه‌سازی مبتنی بر آموزش یادگیری پیشنهاد شده است. در روش پیشنهادی زمان‌بندی بر اساس اطلاعات دریافتی از محیط و ماشین‌های فیزیکی و مجازی صورت می‌گیرد، جهت ارزیابی روش پیشنهادی ازدو معیار محاسبه‌ی کل توان مصرف شده و بررسی نقض کیفیت خدمات استفاده شده است. همانگونه که در نمودارهای فصل پنجم مشاهده می‌شود، روش پیشنهادی MTLBO در مقایسه الگوریتم‌های مشابه با افزایش تعداد ماشین‌های مجازی، نقض SLA کمتری دارد، و با الویت‌بندی و مکان‌یابی ماشین‌های مجازی و استفاده مؤثر از منابع می‌توانیم علاوه بر بهبود کیفیت سرویس، مصرف انرژی را نیز کاهش دهیم. از آنجایی که هر ماشین فیزیکی در حالت کم باری ۶۰٪ زمانی که به صورت کامل مشغول است، مصرف انرژی و توان دارد، به نظر می‌رسد با خاموش کردن ماشین‌های فیزیکی و به حداقل رسانیدن ماشین‌ها می‌توان مصرف انرژی را به حداقل مقدار خود رساند. هدف اصلی الگوریتم بهینه‌ساز MTLBO هم همین است. به عبارت دیگر با خاموش کردن PM ها و همچنین مهاجرت ماشین‌های مجازی به سایر PM ها این عمل رخ داده است.

مراجع:

- [۱] M. Armbrust *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50-58, 2010.
- [۲] A. K. Tripathi, S. Agrawal, and R. D. Gupta, "Cloud enabled SDI architecture: a review," *Earth Science Informatics*, vol. 13, no. 2, pp. 21.۲۰۲۰, ۱-۲۳۱
- [۳] M. Abdullahi, M. A. Ngadi, S. I. Dishing, and S. i. M. Abdulhamid, "An adaptive symbiotic organisms search for constrained task scheduling in cloud computing," *Journal of ambient intelligence and humanized computing*, vol. 14, no. 7, pp. 8.۲۰۲۳, ۸۳۹-۸۸۵.
- [۴] C. L. Albarracín, S. Venkatesan, A. Y. Torres, P. Yáñez-Moretta, and J. C. J. Vargas, "Exploration on Cloud Computing Techniques and Its Energy Concern," *Mathematical Statistician and Engineering Applications*, vol. 72, no. 1, pp. 749-7.۲۰۲۳, ۵۸
- [۵] R. Aggarwal and L. Gupta, "Load Balancing In Cloud Computing," *International Journal Of Computer Science And Mobile Computing*, vol. 6, no. 6, pp. 180-186, 2017.
- [۶] X. Fu, Y. Sun, H. Wang, and H. Li, "Task scheduling of cloud computing based on hybrid particle swarm algorithm and genetic algorithm," *Cluster Computing*, vol. 26, no. 5, pp. 2479-2488, 2023.
- [۷] P. Amini and A. Kalbasi, "An Adaptive Task Scheduling Approach for Cloud Computing Using Deep Reinforcement Learning," in *2024 Third International Conference on Distributed Computing and High Performance Computing (DCHPC)*, 2024, pp. 1-9: IEEE.
- [۸] D. Zhao, G. Sun, D. Liao, S. Xu, and V. Chang, "Mobile-aware service function chain migration in cloud-fog computing," *Future Generation Computer Systems*, vol. 96, pp. 591-604, 2019.
- [۹] E. H. Houssein, A. G. Gad, Y. M. Wazery, and P. N. Suganthan, "Task Scheduling in cloud computing based on meta-heuristics: review, taxonomy, open challenges, and future trends," *Swarm and Evolutionary Computation*, p. 100841, 2021.
- [۱۰] L. Ye, L. Yang, Y. Xia, and X. Zhao, "A Cost-Driven Intelligence Scheduling Approach for Deadline-Constrained IoT Workflow Applications in Cloud Computing," *IEEE Internet of Things Journal*, 2024.
- [۱۱] D. Lakshmi and A. K. Tyagi, "Emerging Technologies and Security in Cloud Computing," 2024.
- [۱۲] C.-H. Youn, M. Chen, and P. Dazzi, "Cost Adaptive Workflow Resource Broker in Cloud," in *Cloud Broker and Cloudlet for Workflow Scheduling*: Springer, 2017, pp. 75-103.
- [۱۳] A. Atadoga, U. J. Umoga, O. A. Lottu, and E. O. Sodiya, "Evaluating the impact of cloud computing on accounting firms: A review of efficiency, scalability, and data security," *Global Journal of Engineering and Technology Advances*, vol. 18, no. 02, pp. 065-074, 2024.
- [۱۴] V. Sharmaa and M. Balab, "Load Balancing in Cloud Computing using Honey Bee Swarm Intelligence," *International Journal of Control and Automation*, vol. 12, no. 6, pp. 428-432, 2019.
- [۱۵] S. Srichandan, T. A. Kumar, and S. Bibhudatta, "Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm," *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 210-230, 2018.
- [۱۶] M. Abd Elaziz, S. Xiong, K. Jayasena, and L. Li, "Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution," *Knowledge-Based Systems*, vol. 169, pp. 39-52, 2019.
- [۱۷] A. Sapkal, L. Heisnam, and S. S. Kusi, "Evolution of Cloud Computing: Milestones, Innovations, and Adoption Trends," 2024.
- [۱۸] I. Attiya, M. Abd Elaziz, and S. Xiong, "Job scheduling in cloud computing using a modified harris hawks optimization and simulated annealing algorithm," *Computational intelligence and neuroscience*, vol. 2020, 2020.
- [۱۹] S. Selvarani and G. S. Sadhasivam, "Improved cost-based algorithm for task scheduling in cloud computing," in *2010 IEEE International Conference on Computational Intelligence and Computing Research*, 2010, pp. 1-5: IEEE.
- [۲۰] M. Mezma *et al.*, "A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems," *Journal of Parallel and Distributed Computing*, vol. 71, no. 11, pp. 1497-1508, 2011.
- [۲۱] C. Mateos, E. Pacini, and C. G. Garino, "An ACO-inspired algorithm for minimizing weighted flowtime in cloud-based parameter sweep experiments," *Advances in Engineering Software*, vol. 56, pp. 38-50, 2013.
- [۲۲] D. Poola, S. K. Garg, R. Buyya, Y. Yang, and K. Ramamohanarao, "Robust scheduling of scientific workflows with deadline and budget constraints in clouds," in *2014 IEEE 28th international conference on advanced information networking and applications*, 2014, pp. 858-865: IEEE.

- [۲۳] A. Allahverdi, "The third comprehensive survey on scheduling problems with setup times/costs," *European Journal of Operational Research*, vol. ۲۴۶, no. 2, pp. 345-378, 2015.
- [۲۴] M. A. Salehi *et al.*, "Stochastic-based robust dynamic resource allocation for independent tasks in a heterogeneous computing system," *Journal of Parallel and Distributed Computing*, vol. 97, pp. 96-111, 2016.
- [۲۵] T. Kumrai, K. Ota, M. Dong, J. Kishigami, and D. K. Sung, "Multiobjective optimization in cloud brokering systems for connected Internet of Things," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 404-413, 2016.
- [۲۶] F. Ebadifard and S. M. Babamir, "A PSO-based task scheduling algorithm improved using a load-balancing technique for the cloud computing environment," *Concurrency and Computation: Practice and Experience*, vol. 30, no. 12, p. e4368, 2018.
- [۲۷] G. Demirci, I. Marincic, and H. Hoffmann, "A divide and conquer algorithm for dag scheduling under power constraints," in *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2018, pp. 466-477: IEEE.
- [۲۸] G. Li, Y. Liu, J. Wu, D. Lin, and S. Zhao, "Methods of resource scheduling based on optimized fuzzy clustering in fog computing," *Sensors*, vol. 19, no. 9, p. 2122, 2019.
- [۲۹] S. P. M. Ziyath and S. Senthilkumar, "MHO: meta heuristic optimization applied task scheduling with load balancing technique for cloud infrastructure services," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-10, 2020.
- [۳۰] A. Al-Moalimi, J. Luo, A. Salah, and K. Li, "Optimal virtual machine placement based on grey wolf optimization," *Electronics*, vol. 8, no. 3, p. 283, 2019.
- [۳۱] N. Patil and D. Aeloor, "A review-different scheduling algorithms in cloud computing environment," in *2017 11th International Conference on Intelligent Systems and Control (ISCO)*, 2017, pp. 182-185: IEEE.
- [۳۲] M. S. R. Krishna and S. Mangalampalli, "A Systematic Review on Various Task Scheduling Algorithms in Cloud Computing," *EAI Endorsed Transactions on Internet of Things*, vol. 10, 2024.
- [۳۳] P. Sen, R. Pandit, and D. Sarddar, "A survey on methods and apparatus of offloading in mobile cloud computing," *International Journal of System of Systems Engineering*, vol. 14, no. 2, pp. 212-225, 2024.

Improving the Scheduling of Static Tasks in Asymmetric Distributed systems Using a Learning-Based Optimization Algorithm

Afshin Shahraki Moghadam

Assistant Professor of Computer Department, Non-Profit Hatef University, Sistan and Baluchistan,
Zahedan

Nima Rigi Karvandar

Master's student in the field of Information Technology Engineering, Non-profit Hatef University,
Sistan and Baluchistan, Zahedan

Abstract

In order to improve the efficiency of distributed systems, it is very important to provide scheduling algorithms. Static task scheduling in heterogeneous distributed environments is an NP-hard optimization problem. The use of meta-revelation algorithms is used to solve these problems. In this research, a static task scheduling algorithm in a new heterogeneous distributed environment based on the optimization algorithm of teaching and learning is proposed. In the proposed algorithm, by considering the priority and execution time of each task and assigning the closest processor to the desired task, the execution time of all tasks is minimized. The convergence of the proposed method is done with a lower number of iterations to reach a relatively optimal solution compared to other meta-revelation algorithms. In order to evaluate, a comprehensive comparison of the evaluation parameters includes agreement of the service level, the amount of energy consumption, the amount of memory used. The evaluation results show that in the proposed method compared to similar methods, the amount of energy consumption by turning off the hosts and physical machines has reached its minimum value compared to other methods. Meanwhile, the agreement of the service level as well as the memory consumption have improved in many evaluation situations.

Keywords: cloud computing, task scheduling, learning-based optimization algorithm, virtual machine migration