

Scheduling tasks in the cloud computing environment using the combination of metal melting algorithm and fuzzy theory

Sajad Manteghi *

Isfahan University, Faculty of Computer
Engineering, Isfahan, Iran.

Roohallah Daneshpayeh

Department of Mathematics, Payame Noor
University, p.o.box. 19395-4697, Tehran, Iran.

Majid Mohammadpour

Yazd University, Faculty of Computer
Engineering, Yazd, Iran.

Abstract

Resource scheduling is one of the most important tasks that is performed in distributed systems such as the cloud environment. On this basis, adopting a suitable method in scheduling can be considered an important matter. The dynamism and heterogeneity of resources in distributed systems causes the complexity of task scheduling. Reducing execution time and execution cost is one of the criteria that is always taken into consideration in all proposed methods for cloud scheduling. Recently, the use of intelligent methods, including the fuzzy system, in the scheduling of tasks in cloud computing has received a lot of attention. Uncertainty and prioritization of input parameters of fuzzy system are important features of fuzzy theory. In this article, a new hybrid scheduling method is presented based on the fuzzy system and the metal melting method, which assigns the requests sent by users to the most suitable source, taking into account criteria such as execution cost, execution time, and imbalance coefficient. The main purpose of the proposed plan is to assign the sent requests to the resources, taking into account the computing power of the resources, the bandwidth of the virtual machines, the delay of the lines between the resources and also the length of the requested work. These parameters are the inputs of the fuzzy system and resources are assigned to requests based on the output of the fuzzy system. The proposed method has been evaluated with CloudSim simulator and the results have been compared with other cloud scheduling methods under the same conditions. The results show that the proposed method improves the efficiency of resource scheduling in terms of total execution time, execution cost and imbalance coefficient.

Keywords: cloud computing, scheduling, fuzzy theory, virtual machine, metal melting

Received: 29/January/2023

Accepted: 08/May/2023

ISSN: 2980-8936

زمان‌بندی کارها در محیط محاسبات ابری با استفاده از ترکیب الگوریتم ذوب فلزات و تئوری فازی

سجاد منطقی *

دانشگاه اصفهان، دانشکده مهندسی کامپیوتر، اصفهان، ایران.

روح‌الله دانش‌پایه

استادیار گروه ریاضی، دانشگاه پیام نور، تهران، ایران.

مجید محمدپور

دانشجوی دکتری، دانشگاه یزد، دانشکده مهندسی کامپیوتر، یزد، ایران.

چکیده

زمان‌بندی منابع، یکی از مهم‌ترین کارهایی است که در سیستم‌های توزیع‌شده نظیر محیط ابری انجام می‌شود. بر همین مبنا اتخاذ روشی مناسب در زمان‌بندی می‌تواند امری مهم تلقی گردد. پویایی و ناهمگونی منابع در سیستم‌های توزیع‌شده باعث پیچیدگی زمان‌بندی وظایف می‌شود. کاهش زمان اجرا و هزینه اجرا از معیارهایی است که همواره در تمامی روش‌های پیشنهادی برای زمان‌بندی ابر، مورد توجه قرار می‌گیرد. اخیراً استفاده از روش‌های هوشمند از جمله سیستم فازی در زمان‌بندی کارها در محاسبات ابری مورد توجه بسیاری قرار گرفته است. عدم قطعیت و اولویت‌بندی پارامترهای ورودی سیستم فازی از ویژگی‌های مهم تئوری فازی است. در این مقاله، یک روش زمان‌بندی ترکیبی جدید بر پایه سیستم فازی و روش ذوب فلزات ارائه شده است که درخواست‌های ارسالی از طرف کاربران را با در نظر گرفتن معیارهایی هم‌چون هزینه اجرا، زمان اجرا و ضریب عدم تعادل به مناسب‌ترین منبع تخصیص می‌دهد. هدف اصلی طرح پیشنهادی اختصاص دادن درخواست‌های ارسالی به منابع با در نظر گرفتن توان محاسباتی منابع، پهنای باند ماشین‌های مجازی، میزان تأخیر خطوط بین منابع و هم‌چنین طول کار درخواستی می‌باشد. این پارامترها، ورودی‌های سیستم فازی هستند و بر اساس خروجی سیستم فازی منابع به درخواست‌ها اختصاص داده می‌شوند. روش پیشنهادی با شبیه‌ساز کلودسیم ارزیابی شده و نتایج با روش‌های دیگر زمان‌بندی ابر تحت شرایط یکسان مقایسه شده است. نتایج نشان می‌دهد که روش پیشنهادی کارایی زمان‌بندی منبع را در معیارهای کل زمان اجرا، هزینه اجرا و ضریب عدم تعادل بهبود می‌بخشد.

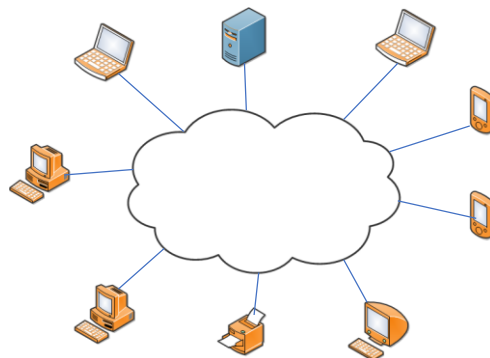
کلیدواژه‌ها: محاسبات ابری، زمان‌بندی، تئوری فازی، ماشین مجازی، ذوب فلزات

۱- مقدمه

یکی از مهم‌ترین کارهایی که در سیستم‌های توزیع شده نظیر محیط ابر انجام می‌شود زمان‌بندی درخواست‌های ارسالی از طرف کاربران است. بر همین مبنا اولویت‌دهی معیارهای زمان‌بندی نظیر توان محاسباتی منابع و طول کار ارسالی کاربر نیز می‌تواند امری مهم تلقی گردد. پویایی و ناهمگونی منابع در سیستم‌های توزیع شده باعث پیچیدگی زمان‌بندی وظایف می‌شوند. اکثر سیستم‌های زمان‌بندی موجود، سعی دارند زمان اتمام وظیفه و هزینه اجرای وظیفه‌ها را به صورت مجزا بهینه نمایند.

نمونه‌های متنوعی از سیستم‌های محاسباتی ارائه شده است که سعی دارند خدمات زمان‌بندی را به کاربران ارائه دهند. برخی از این سیستم‌های محاسباتی عبارت‌اند از: محاسبات کلاستری^۱ محاسبات توری^۲ و اخیراً محاسبات ابری که از آن به عنوان محاسبات انبوه^۳ نیز یاد می‌شود. محبوبیت این سه رویکرد محاسباتی، از دید موتور جستجوی گوگل مورد ارزیابی قرار گرفته است که نتیجه آن حاکی از آن است که محبوبیت محاسبات ابری، پس از ظهور مفاهیم اولیه آن در سال ۲۰۰۷، با فاصله زیادی نسبت به سایر رویکردهای محاسباتی در حال افزایش است (اکبری و سرگلزایی جوان، ۱۳۸۹). دنیای محاسباتی به سرعت به سمت توسعه نرم‌افزارهایی پیش می‌رود که به جای اجرا بر روی رایانه‌های منفرد، به عنوان یک سرویس در دسترس میلیون‌ها مصرف‌کننده قرار داده می‌شوند. از این نقطه نظر، محاسبات انبوه (محاسبات ابری) از دید کاربران نهایی ساختاری شبیه به یک توده ابر دارد که به واسطه آن می‌توانند به برنامه‌های کاربردی از هر جایی از دنیا دسترسی داشته باشند.

اما محاسبات انبوه از دید فراهم‌کنندگان منابع زیرساخت، می‌تواند با کمک ماشین‌های مجازی^۱ شبکه شده، به عنوان یک روش جدید برای ایجاد پویای نسل جدید مراکز داده، مورد استفاده قرار گیرد تا بتوانند یک زیرساخت قابل انعطاف برای ارائه انواع مختلف خدمات محاسباتی و ذخیره‌سازی در اختیار داشته باشند. در رویکرد اول (محاسبات ابری) از دید ارائه سرویس و برنامه‌های کاربردی تلاش بر این است که خدمات اینترنتی به صورت یک رایانه واحد در اختیار تمام کاربرانی که به آن متصل هستند قرار بگیرد و تکنولوژی‌هایی نظیر وب^۲ از عوامل مهم در نیل به این هدف هستند. در این رویکرد جدید، می‌توان از لایه‌های مختلف و قابل انعطاف ارائه شده در ابر استفاده کرد و خدماتی را فراهم آورد که بتوان در سایت‌های مختلف به اشتراک گذاشت. به این ترتیب که مثلاً اطلاعات می‌تواند بین سایت‌ها به اشتراک گذاشته شود و فایلی که در یک سایت قرار داده شده است به راحتی در یک سایت دیگر قابل دسترسی باشد. در حقیقت حرکت در ابر به سمتی پیش می‌رود که دیگر برای کاربر فرقی نمی‌کند در حال استفاده از کدام سایت است. بلکه کاربر کل اینترنت را همانند یک رایانه شخصی در خدمت خود مشاهده می‌کند (شکل ۱). مصداق‌های بسیار زیادی از این روند در حال حاضر قابل مشاهده است. برای مثال کاربر می‌تواند کلیپی را در سایت Youtube مشاهده نموده و آن را در یک سایت دیگر مثلاً Facebook به اشتراک بگذارد.



شکل (۱): یکپارچه‌سازی کلیه خدمات ارائه‌شده اینترنت از دید کاربر

کاملاً واضح است که در چنین محیطی کار با سایت‌های مختلف، همانند کار با برنامه‌های کاربردی مختلف در یک رایانه شخصی می‌باشد و گرایش به سمتی است که دیگر کاربر مرزی را بین این خدمات احساس نکند و بتواند با همان درجه آزادی که در رایانه شخصی خود دارد، در اینترنت نیز به فعالیت بپردازد؛ اما اگر بخواهیم به چنین سطحی از انعطاف‌پذیری در ارائه خدمات کاربردی به کاربران دست پیدا کنیم، نیاز به زیرساختی قابل‌انعطاف خواهیم داشت که بر اساس رویکرد دوم و توسط فراهم‌کنندگان منابع زیرساخت می‌توان به آن دست پیدا کرد.

هدف تعادل بار در محاسبات ابری، توزیع مساوی به هر منبع است به طوری که بیش‌ترین کارآیی از هر منبع گرفته شود و کم‌ترین زمان اجرا برای هر کار تخصیص داده شود (LD & Krishna, 2013). در محاسبات ابری، هدف این است که مدیریت سیستم به نحوی انجام گیرد که مجموعه کارهای موردنظر در کوتاه‌ترین زمان ممکن انجام شود. یکی از مواردی که باعث ارتقای کارآیی و کاهش زمان اجرای کار می‌شود، زمان‌بندی کارها می‌باشد. منظور از زمان‌بندی کارها این است که مشخص شود که هر کار در چه زمانی و در چه ماشینی اجرا شود. در این مقاله قصد بر این است که به کمک سیستم فازی با توجه به پارامترهای مطرح‌شده مانند طول کار وارد شده به سیستم و مؤلفه‌های منابع عملیات زمان‌بندی به نحوی انجام گیرد که در نهایت منجر به افزایش بهره‌وری، کاهش زمان اجرا و بهبود توازن بار گردد. طول کار وارد شده به سیستم و مؤلفه‌های منابع (توان محاسباتی منابع، پهنای باند خطوط ارتباطی منابع و میزان توان محاسباتی مصرف‌شده در منابع) پارامترهای ورودی سیستم فازی هستند و بر مبنای خروجی سیستم فازی، زمان‌بندی انجام می‌گیرد.

زمان‌بندی منبع در محیط ابری به معنی اتخاذ استراتژی‌هایی هست که سبب گردد حداکثر استفاده بهینه از منابع محیط ابر حاصل شود. ارائه یک راه کار جدید در زمان‌بندی منبع، به عنوان اصلی‌ترین چالش و هدف این مقاله است. در حقیقت هدف این مقاله، ارائه یک چارچوب هوشمند است که با کمک ترکیب روش الگوریتم فرااکتشافی ذوب فلزات و سیستم هوشمندی هم‌چون سیستم فازی با حداقل متوسط زمان خاتمه کارها، زمان‌بندی کارها صورت گیرد. بهبود در متوسط زمان خاتمه کارها، از ویژگی‌های چارچوب پیشنهادی خواهد بود. سؤال اصلی که طی اجرای این تحقیق پاسخ داده شده است عبارت است از "چگونه می‌توان یک چارچوب کلی برای زمان‌بندی منبع را با استفاده از معماری محیط ابری و پروتکل‌های مورد استفاده در این نوع شبکه با کمک سیستم‌های هوشمند ارائه نمود به گونه‌ای که در عین حال کیفیت خدمات نیز تأمین شود؟" فن‌آوری مربوط به محاسبات ابری در دهه اخیر پیشرفت‌های قابل توجهی را در محاسبات علمی داشته است. این فناوری تعداد زیادی کامپیوترهای اتصال داخلی را یکی کرده و اشتراک منابع را شبیه به آنچه در اینترنت داریم فراهم می‌نماید.

ساختار این مقاله به صورت زیر می‌باشد:

در بخش اول این مقاله مقدمه و کلیات پژوهش بیان می‌شود. در بخش دوم مروری بر کارهای گذشته صورت می‌گیرد. بخش سوم این مقاله روش پیشنهادی را تشریح می‌نماید. بخش چهارم این مقاله نتایج تجربی و آزمایش‌ها را برای روش پیشنهادی بیان می‌کند. در نهایت بخش پنجم این مقاله شامل نتیجه‌گیری و پیشنهادها آتی می‌باشد.

۲- کارهای گذشته

در این بخش به معرفی برخی الگوریتم‌های زمان‌بندی پرداخته می‌شود که این الگوریتم‌ها در بخش‌های مختلف محیط ابر استفاده می‌شوند.

چن و همکاران^۱ (۲۰۱۰)، الگوریتم و چارچوبی برای زمان‌بندی پردازش‌ها ارائه کرده‌اند. این الگوریتم، سرویس‌دهی به درخواست‌ها را به وسیله مدیریت کردن منابع موجود به صورت سلسله‌مراتبی مدیریت می‌کند. نویسندگان درون معماری ابر بر روی سرویس‌های SaaS، یک مدل به نام ناحیه بافر^۲ ارائه کردند که کار آن تقسیم وظایف درخواستی کاربر به مراحل مختلف است. کار هر مرحله ایجاد متدهای مختلف و ارسال آن‌ها به منابع موجود در ابر جهت پردازش درخواست‌ها است. در آخر نیز کیفیت نتایج بازگشتی بررسی می‌شود. الگوریتم ارائه شده دارای یک الگوریتم کمکی است. این الگوریتم، هنگامی که کیفیت نتایج مطلوب نبود، دو مرحله آخر را دوباره انجام می‌دهد. با این روش از دوباره اجرا کردن وظایف جلوگیری به عمل می‌آید.

پاول و سانیا^۳ (۲۰۱۱)، روشی برای اختصاص درخواست‌ها به منابع مناسب، با هدف به حداقل رساندن زمان اتمام کل اجرا ارائه کرده‌اند. در این روش، از روی اعتبار استفاده شده هر وظیفه که به یک منبع اختصاص داده شده، یک ماتریس هزینه ایجاد می‌شود. هر وظیفه با داشتن اعتباری بیشتر دارای شانس بیشتری برای اختصاص یافتن به بهترین منابع موجود می‌باشد. روش پیشنهادی برای محیط ابر همگن ارائه شده است. در این روش یک معماری ارائه شده است که وظیفه‌های ارسالی کاربران را در بافر یک میان‌افزار مرکزی^۴ ذخیره می‌کند. سپس وظیفه‌ها را به صورت سلسله‌مراتبی بین میان‌افزارهای محلی^۵ تقسیم می‌کند. بعد از این تقسیم‌بندی، زمان‌بندی توسط میان‌افزارهای محلی انجام می‌شود.

لین و همکاران^۶ (۲۰۰۹) از محاسبات ابری به عنوان یک مفهوم جدید در تکنولوژی و تجارت یاد کرده‌اند که برای افراد مختلف معانی مختلفی دارد. برای کاربران برنامه‌های کاربردی به مفهوم ITaaS^۷ شناخته می‌شود که امکان ذخیره‌سازی، محاسبه و استفاده از برنامه‌های کاربردی را از طریق اینترنت فراهم می‌کند. برای توسعه‌دهندگان وب یک سکوی توسعه نرم‌افزار و محیط اجرای نرم‌افزار در مقیاس اینترنت به حساب می‌آید؛ و برای مدیران و فراهم‌کنندگان ارتباطات زیرساخت، مانند یک زیرساخت مبتنی بر مراکز داده‌ای توزیع شده و عظیم می‌باشد که از طریق شبکه به هم متصل هستند.

یکی از تکنولوژی‌های مورد استفاده در محاسبات ابری، مجازی‌سازی است (Liu et al., 2014). مجازی‌سازی این امکان را فراهم می‌آورد که بتوان در یک رایانه یا سرور به طور هم‌زمان چندین سیستم عامل، یا چندین بخش مختلف از یک سیستم عامل را اجرا کرد. هم‌چنین باعث تسهیل مقیاس‌پذیری و انعطاف‌پذیری در ایجاد و سرهم کردن منابع مورد نیاز مشتری می‌شود که ممکن است به طور موقت یا در شرایط خاص (مثلاً در زمان‌های اوج مصرف) انجام

1. Chen, M., Li, M., & Cai, F.

2. Buffer - pool

3. Paul, M., & Sanyal, G.

4. Central middleware

5. Local middleware

6. Lin, G., Fu, D., Zhu, J., & Dasmalchi, G.

7. IT as a Service

شود. یک ایستگاه مجازی شکلی انتزاعی و ایزوله‌شده از یک محیط اجرا است که می‌تواند به‌صورت پویا و از طریق پروتکل‌های مشخص در دسترس یک کاربر مجاز قرار بگیرد. در این ایستگاه مجازی می‌توان منابعی نظیر میزان پردازنده و یا حافظه را به شکل دلخواه اختصاص داد و یا تنظیمات نرم‌افزاری مشخصی را انجام داد (Wu et al., 2012). هر ایستگاه مجازی تحت عنوان یک تصویر ۱ قابل‌ایجاد و مدیریت است.

لی و همکاران^۱ (۲۰۱۱) الگوریتمی به نام PISA^۲ ارائه کردند که در آن به زمان‌بندی وظیفه‌ها با اولویت‌های درخواستی کاربر پرداخته‌اند. مشکلی که در این مقاله مطرح شده، حجم متفاوت وظیفه‌ها و برقراری تعادل بین اولویت‌دهی‌های سیستمی و اولویت‌های درخواستی کاربران می‌باشد. راه‌حل پیشنهادی برای این مشکل، به این صورت است که بعد از هر بار تخصیص منبع، وظیفه‌ها دارای یک اولویت بیشتری شوند؛ در صورتی که بعد از چند دور اگر اولویت‌ها به مقدار ثابت رسید، دیگر افزایش نداشته باشد. نتایج به‌دست آمده نشان می‌دهد که میزان موفقیت در این روش از روش FIFO بیشتر می‌باشد؛ زیرا علاوه بر گردش کار، وزن نیز تأثیر دارد.

جین و همکاران^۳ (۲۰۱۱) یک چارچوب برای برنامه‌های کاهش نگاشت^۴ به‌منظور کاهش انرژی برای انجام محاسبات بدون این که از درخواست‌های کاربران چیزی را نقض کند، ارائه کردند. چارچوب پیشنهادی با استفاده از اطلاعاتی در مورد توزیع کلید میانی برای انتخاب پردازنده مناسب برای کاهش وظایف، باعث صرفه‌جویی در توان پردازشی در محیط ابر خواهد شد.

سوگو^۵ (۱۹۸۵) یک الگوریتم زمان‌بندی اکتشافی به نام BAR^۶ ارائه کرده‌اند که الگوریتم شامل دو فاز تعادل و کاهش می‌باشد. در فاز تعادل همه وظیفه‌ها به منابع مشخص تخصیص داده شده‌اند. در این حالت هر منبع (سرور) وظیفه‌های داخلی خود را دارد، اما به دلیل اینکه وظیفه‌ها به هم وابسته هستند، برای اجرا هر وظیفه نیاز به فراخوانی وظیفه‌های پدر از منابع دیگر می‌باشد. در فاز کاهش، برای رعایت کردن اصل محلیت^۷، هر منبع، وظیفه‌های موردنظر را از منابع مختلف جمع‌آوری می‌کند یا بهتر است بگوییم وظیفه‌ها روی ماشین‌های مختلف تکرار می‌شوند.

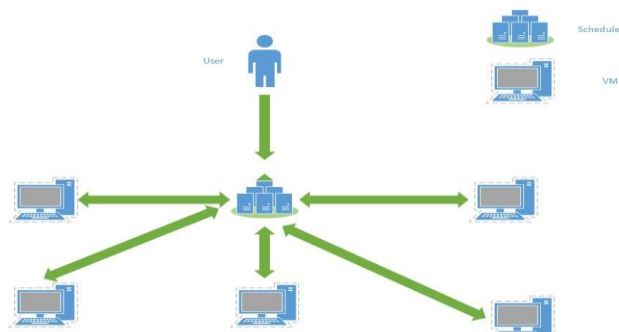
۳- روش پیشنهادی

در طرح پیشنهادی، برای تخصیص منابع به کارها، یک الگوریتم زمان‌بندی توزیع‌شده با هدف توازن بار ارائه شده است. این الگوریتم با حفظ توازن بار، مانع از نادیده گرفتن ناعادلانه منابع با توان پردازشی بالا هنگام تخصیص منبع به کار می‌شود. در ضمن میزان طول کارها را نیز در نظر می‌گیرد. دلیل اصلی استفاده از سیستم فازی، عدم قطعیتی است که به طرح می‌دهد و در ضمن چهار پارامتر یادشده هم‌زمان در نظر گرفته می‌شوند. میزان کارایی سیستم فازی با روش ممدانی^۸ بستگی زیادی به قوانین فازی تعریف شده دارد. طبق محیط‌های مختلف می‌توان با تعریف قوانین فازی مناسب میزان تأثیرگذاری پارامترها را کم و زیاد کرد و خاصیت پویایی به روش پیشنهادی داد.

وقتی زمان‌بند، یک کار ثبت‌شده را از سوی کاربر دریافت می‌کند، به کمک یک کنترل‌کننده ابر (Cloud controller)، کارها را به ماشین‌های مجازی انتساب می‌دهد. در روش پیشنهادی از روش توزیع‌شده استفاده شده است که در آن عامل زمان‌بند در کنترل‌کننده ابر قرار داده شده است. (عامل یک برنامه است که متناسب با کاری که انجام می‌دهد به همان نام نام‌گذاری می‌شود). کنترل‌کننده ابر، میزان مناسب بودن کار دریافتی را با در نظر گرفتن

1. Li, Y., Zhang, H., & Kim, K. H.
2. Priority Impact Scheduling Algorithm
3. Jin, J., Luo, J., Song, A., Dong, F., & Xiong, R.
4. MapReduce
5. Sugeno
6. Balance-Reduce
7. Locality
8. Mamdani

قابلیت ماشین‌های مجازی، با استفاده از سیستم زمان‌بندی پیشنهادی محاسبه می‌کند. کنترل‌کننده ابر طبق اعدادی که حاصل از انجام مقایسه بین نودها است، زمان‌بندی را انجام می‌دهد. کنترل‌کننده ابر، پروکر شبکه ابری است که لیست دیتاست‌ها و سرورها را در اختیار دارد. شکل ۲ چارچوب سیستم زمان‌بندی طرح پیشنهادی را نشان می‌دهد.



شکل (۲): چارچوب سیستم زمان‌بندی طرح پیشنهادی

فرآیند زمان‌بندی، وظایف را به منابع مناسب هدایت می‌کند و این کار باید به گونه‌ای صورت گیرد که بار کاری به صورت متوازن روی منابع توزیع شود تا حداکثر بهره از منابع موجود حاصل شود. ایجاد توازن بار یکی از شاخص‌های مهم کارایی مدیریت منابع در محاسبات ابری است. در زمان‌بندی طرح پیشنهادی، کارها و هم‌چنین منابع، دارای اولویت هستند که اولویت‌شان برای منابع طبق مؤلفه‌های ماشین‌های مجازی و برای کارها طبق طول کار توسط سیاست‌های محیط ابر توسط مدیر سیستم تعیین می‌گردد.

هر فضای جستجو^۱ با نماد S نشان داده می‌شود. هر کدام از این حالت‌ها جوابی می‌تواند برای مسئله زمان‌بندی تلقی شود. در محیط ابری، هر S بیانگر یک کار^۲ است که به یک ماشین مجازی^۳ تخصیص داده شده است. الگوریتم از یک وضعیت اولیه شروع می‌شود و به سمت وضعیت‌های همسایه پیش می‌رود. در هر دور تکرار، وضعیت فعلی با وضعیت همسایه مجاور مقایسه می‌شود و در صورتی که وضعیت همسایه از وضعیت فضای جستجوی فعلی بهتر باشد وضعیت همسایه به عنوان جواب برای کار وارد شده در نظر گرفته می‌شود. به عبارت ساده‌تر کار وارد شده برای تخصیص به ماشین مجازی که در وضعیت فعلی قرار دارد در نظر گرفته می‌شود و سپس همین کار با ماشین مجازی که در S همسایه مجاور قرار دارد مقایسه می‌شود. چنانچه ماشین مجازی وضعیت همسایه بهتر از وضعیت فعلی باشد کار وارد شده برای اجرا به ماشین مجازی وضعیت همسایه در نظر گرفته می‌شود. الگوریتم پیشنهادی میزان مناسب بودن این دو وضعیت را با منطق فازی انجام می‌دهد. سیستم طرح پیشنهادی از تئوری فازی استفاده می‌کند که چهار معیار توان محاسباتی منابع، طول کار وارد شده به سیستم، پهنای باند و میزان حافظه پارامترهای ورودی آن هستند. هر چهار معیار، درون سه بازه کم، متوسط و زیاد قرار دارند که این بازه‌ها با یکدیگر هم‌پوشانی دارند. در طرح پیشنهادی از مدل فازی برای زمان‌بندی استفاده می‌شود. مدل پیشنهادی بر اساس منطق فازی است که مقادیری را به عنوان ورودی دریافت می‌کند. این مقادیر پارامترهای ورودی سیستم فازی هستند. این اعداد غیر فازی به سیستم استنتاج فازی فرستاده می‌شوند و به وسیله قواعد فازی و استدلال فازی منبع مناسب انتخاب می‌شوند و در قالب عددی غیر فازی نشان داده می‌شوند. سیستم فازی استفاده شده در مرکز استنتاج در این مقاله، سیستم فازی ممدانی است. سیستم فازی ممدانی یک

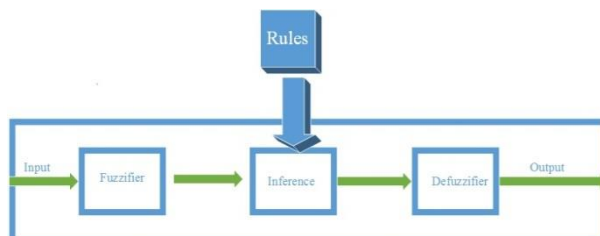
1. Searching space

2. Job (Task)

3. Virtual machine (VM)

روش ساده قانون-مبنا^۱ است که محاسبات پیچیده‌ای نیاز ندارد و می‌تواند سیستم‌ها را با استفاده از قوانین اگر... آنگاه... کنترل کند.

شکل ۳ ساختار موتور استنتاج فازی طرح پیشنهادی را نشان می‌دهد.



شکل (۳): ساختار موتور استنتاج فازی طرح پیشنهادی

دو نوع متداول سیستم استنتاج فازی وجود دارد (Sugeno, 1985): نوع Mamdani و Sugeno. سیستم استنتاج فازی شامل ۵ مرحله است. فازی‌سازی متغیرهای ورودی، اعمال عملگرهای فازی، اعمال روش دلالت، اجتماع تمام خروجی‌ها و مرحله غیرفازی کردن ورودی‌ها. قدم اول این است که ورودی‌ها را دریافت کرده و برای هر کدام از ورودی‌ها توسط توابع عضویت مربوط به مجموعه فازی مناسب، یک درجه به آن‌ها نسبت داده شود. برای این منظور سه مجموعه فازی تشکیل داده می‌شود که با یکدیگر هم پوشانی دارند. از محدوده ۰ تا ۰٫۵، در محدوده فاصله کم^۲، محدوده ۰ تا ۱ در محدوده فاصله متوسط^۳ و محدوده ۰٫۵ تا ۱ در محدوده فاصله زیاد^۴ قرار دارد. بهتر است بازه‌ها به صورتی انتخاب شوند که نقاط ابتدا و انتهای دو بازه‌ای که با یک فاصله از یکدیگر قرار دارند مثلاً بازه کم و زیاد، بر هم منطبق شوند. با استفاده از تابع عضویت گروهی که گره باید درون آن قرار گیرد مشخص می‌گردد.

توابع عضویت گوناگونی وجود دارد از جمله مثلثی، دوزنقه‌ای، گوسین، محدب و غیره. در این تحقیق از توابع عضویت مثلثی استفاده می‌شود. استفاده از این توابع علاوه بر داشتن دقت کافی، نیاز به پردازش سنگینی ندارند و برای سیستم‌های توزیع شده مناسب هستند. یک تابع عضویت در حقیقت، منحنی‌ای است که نشان می‌دهد، یک نقطه در فضای ورودی چگونه به مقدار عضویت در خروجی نگاشت می‌شود. μ درجه عضویت را نشان می‌دهد، که عددی بین ۰ تا ۱ است. به طور کلی رابطه (۱) برقرار است:

$$\mu_A(x) = \text{Degree}(x) \text{ in } A$$

$$\forall x \in X: \mu_A(x): X \rightarrow [0,1] \quad (1)$$

در طرح پیشنهادی به ازای هر گره مجازی تعداد ۸۱ قانون فازی می‌تواند وجود داشته باشد (سه بازه وجود دارد؛ پس عدد پایه، ۳ است و چهار پارامتر وجود دارد که توان آن هم ۴ می‌شود) که باید طبق روش‌های بهینه‌سازی قوانین فازی این قوانین را بهینه کرد و از تعداد آن‌ها کاست.

یکی از ساده‌ترین راه‌ها در روش ممدانی استفاده از تجربیات گذشته است که این تجربیات گذشته به صورت فرضیه در مورد استفاده از قوانین در نظر گرفته می‌شود؛ یعنی با توجه به معیارهای استفاده شده و تجربیات گذشته در محیط‌های واقعی میزان اولویت و اهمیت معیارها تعریف می‌شود. دقت شود که در قوانین فازی تعداد قوانین مطرح نیست و نتیجه قانون، کارآیی قوانین فازی را معلوم می‌کند. بدین صورت که نتیجه مناسب در ستون نتیجه قوانین بدین معنی است که کار وارد شده برای ماشین مجازی مدنظر مناسب است و نتیجه نامناسب بدین معنی است که کار وارد شده برای ماشین مجازی مدنظر مناسب نیست. در جدول ۱ قوانین بهینه شده طرح پیشنهادی نشان داده شده است که حاصل

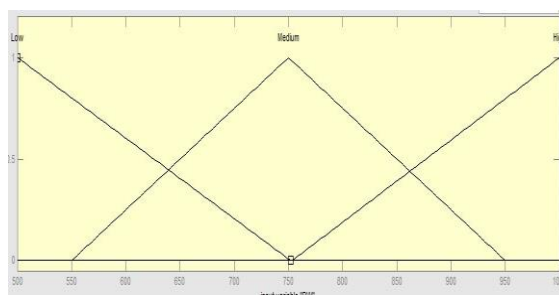
1. rule-base
2. Low distance
3. Medium distance
4. High distance

تجربیات گذشته در محیط‌های واقعی کلود است. در فرآیند زمان‌بندی، چهار پارامتر پهنای باند^۱، میزان توان محاسباتی نودها^۲، حافظه ماشین مجازی^۳ و طول کار^۴ به عنوان ورودی سیستم فازی در نظر گرفته شده است.

جدول (۱): قوانین بهینه استفاده شده در سیستم استنتاج فازی

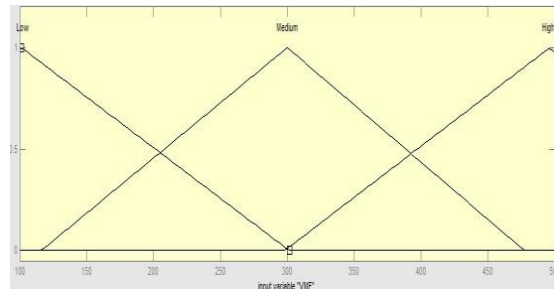
rule	Bandwidth (BW)	VM frequency (VMF)	VM memory (VMM)	Job length (JL)	Result
1	Low	Low	Low	Low	Adequate
2	Low	High	Low	Medium	Medium
3	Medium	Medium	Low	Medium	Adequate
4	High	High	Medium	Medium	Adequate
5	High	High	Low	High	Adequate
6	Low	Medium	Medium	Low	Medium
7	Low	Low	Medium	Medium	Medium
8	Low	Low	Medium	Low	Adequate
9	Low	High	High	High	Medium
10	Low	Low	High	High	Inappropriate
11	Medium	Medium	High	High	Inappropriate
12	High	Low	High	High	Medium
13	Medium	Medium	High	Low	Inappropriate
14	Medium	High	High	Low	Inappropriate
15	Low	High	High	Medium	Medium
16-81					Not used

شکل ۴ بازه پارامتر پهنای باند، شکل ۵ بازه پارامتر توان محاسباتی نودها، شکل ۶ بازه پارامتر مقدار مصرف شده پردازنده و شکل ۷ بازه پارامتر طول کار وارد شده را نشان می‌دهند.

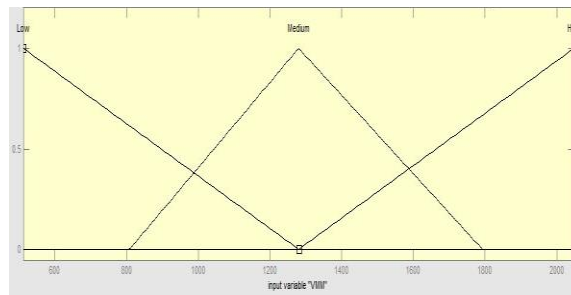


شکل (۴): بازه پارامتر پهنای باند

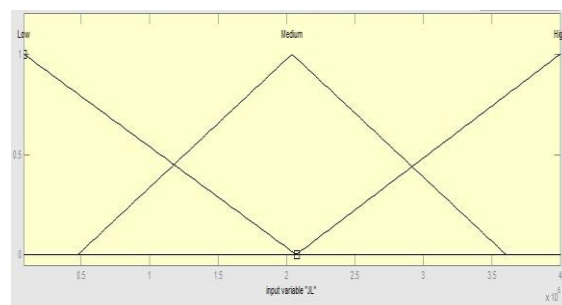
1. Bandwidth (BW)
2. Node Power (NP)
3. VM memory (VMM)
4. Job length (JL)



شکل (۵): بازه پارامتر توان محاسباتی نودها



شکل (۶): حافظه ماشین مجازی



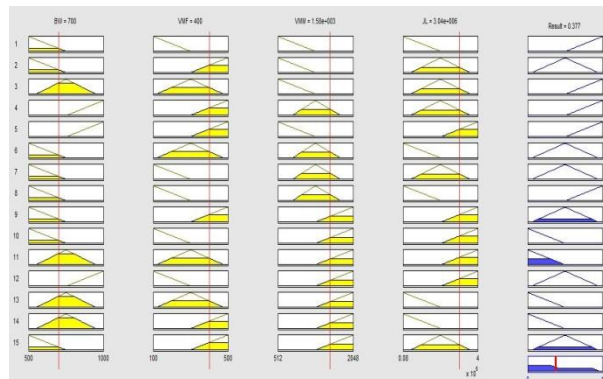
شکل (۷): بازه پارامتر طول کار واردشده

عملگرهای AND و OR دو عملگر استفاده شده رایج در مجموعه قوانین فازی هستند. AND به معنای کمینه و OR به معنای بیشینه است. بعد از اعمال عملگرهای فازی روش دلالت را باید اعمال نمود. ورودی فرآیند دلالت یک عدد است که از مقدم به دست آمده می‌آید و خروجی آن یک عدد فازی است. دلالت برای هر کدام از قوانین اجرا می‌شود که شامل کمینه و حاصل ضرب است. بعد از این مرحله، اجتماع تمام قوانینی که به اصطلاح fire شده‌اند را باید محاسبه کرد. مرحله آخر، مرحله غیرفازی کردن است. غیرفازی کردن یعنی تبدیل یک نمودار فازی به یک عدد. در واقع استخراج عددی که بتواند گویای آن مجموعه فازی باشد. ۵ روش برای غیرفازی‌سازی وجود دارد: مرکز ثقل، نیم‌ساز، نصف ماکزیمم، بزرگ‌ترین ماکزیمم و کوچک‌ترین ماکزیمم. متداول‌ترین روش از بین این روش‌ها غیرفازی‌سازی روش محاسبه مرکز ثقل است که به وسیله عملگر مرکز ثقل^۱ پیاده‌سازی می‌شود. در این مقاله از روش مرکز ثقل برای عکس فازی‌سازی استفاده شده است. محاسبه مرکز ثقل یعنی میانگین وزن دار هر نقطه روی دامنه به طوری که وزن هر نقطه مقدار درجه عضویت آن است.

محاسبه مرکز ثقل چیزی شبیه به محاسبه معدل درسی است که برای قوانین از شماره ۱ تا N انجام می‌شود. μ شبیه به تعداد واحد درس است و در واقع درجه عضویت قانون fire شده است. رابطه (۲) نشان‌دهنده روش مرکز ثقل است.

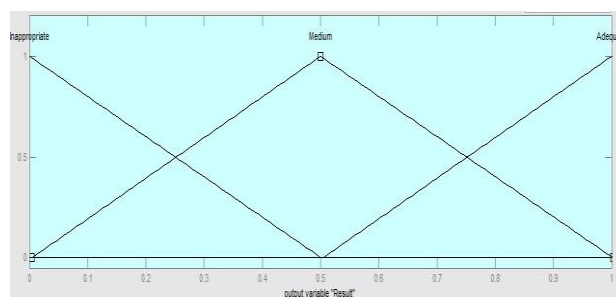
$$Y = \frac{\sum y_i \cdot \mu(y_i)}{\sum \mu(y_i)} \quad (2)$$

شکل ۸ اجتماع خروجی قوانینی که به کار بسته شده‌اند^۱ و هم‌چنین مرحله غیرفازی سازی به وسیله مرکز ثقل را که به وسیله قسمت نمایشگر قانون نرم افزار متلب کشیده شده است را نشان می‌دهد.



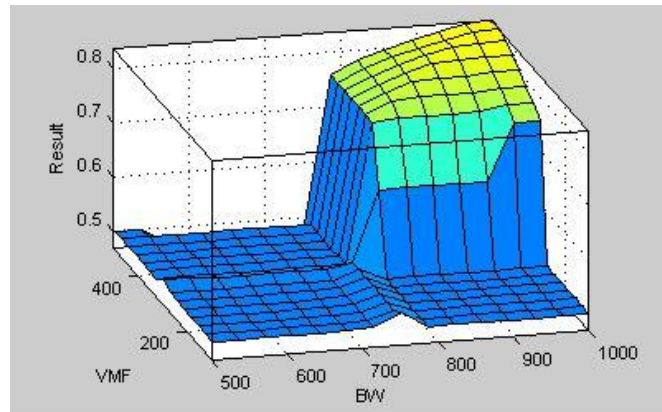
شکل (۸): اجتماع خروجی قوانین فازی و مرحله غیرفازی سازی به وسیله مرکز ثقل

مقدار حاصل از رابطه (۲) تخصیص یا عدم تخصیص به ماشین مجازی را تعیین می‌کند. طبق این رابطه ماشین مجازی که بیشترین مقدار را از خروجی فازی داشته باشد برای کار وارد شده انتخاب می‌گردد. در شکل ۹ بازه خروجی پارامتر فازی نشان داده شده است.



شکل (۹): بازه خروجی پارامتر فازی

در نهایت نمایشگر قانون اجازه می‌دهد تا فرآیند استنتاج فازی یک‌جا تفسیر شود. هم‌چنین این قسمت نشان می‌دهد چگونه قوانینی که به کار بسته شده‌اند در جواب نهایی تأثیر دارند. برای مشاهده مجموعه‌های خروجی برحسب مجموعه‌های ورودی از قسمت نمایشگر سطح استفاده می‌شود که در شکل ۱۰ نشان داده شده است. نمایشگر سطح، شکلی سه‌بعدی از ورودی‌ها و خروجی‌ها در مرحله زمان‌بندی منابع را نشان می‌دهد. طبق خروجی فازی، چنانچه این مقدار برای ماشین مجازی وضعیت همسایه از ماشین مجازی وضعیت فعلی بیشتر باشد ماشین مجازی وضعیت همسایه را برای تخصیص به کار وارد شده در نظر می‌گیرد. در هر مقایسه، یک واحد از مقدار متغیر دما که با نماد T نشان داده شده کم می‌شود.



شکل (۱۰): قسمت نمایشگر سطح که مجموعه‌های خروجی بر حسب مجموعه‌های ورودی در قسمت فازی را نشان می‌دهد.

الگوریتم تا زمانی که مقدار متغیر دما به صفر نرسد، ادامه پیدا می‌کند. برای کاهش متغیر دما این متغیر را تابعی بر حسب ماشین‌های مجازی در دسترس انتخاب می‌کنیم تا دسترس‌پذیری طرح را هم در نظر گرفته باشیم. برای روشن‌تر شدن طرح پیشنهادی، شبه‌کد طرح پیشنهادی در زیر آورده شده است.

```

FSACS Pseudo Code
t=T0
Initial Solution= S0
Best solution= S0
OF0= Object Function(S0)
OF Best Solution= OF0
For i=1 to Max do {
S1= Generate neighbor(S0)
OF1= Object Function(S1) //this object function is done
based on fuzzy theory
If OF1 ≥ OF0 then {
S0= S1
OF0 = OF1
If OF1 ≥ OF Best Solution then {Best Solution= S1
OF Best Solution= OF1 }
}
Else {
If QOS[S1] > QOS[s0] {
S0= S1
OF0 = OF1}
}
T=0.9 * t
If t=0 {
Return (Best Solution)
Halt()}//end for
Return (Best Solution)

```

۴- شبیه‌سازی و نتایج تجربی

این بخش، به بیان نتایج به‌دست آمده از شبیه‌سازی‌های انجام شده می‌پردازد. در این آزمایش‌ها الگوریتم پیشنهادی، در معیارهای مختلف ارزیابی با پارامترهای مختلف مربوط به منابع با الگوریتم‌هایی مشابه، در محیط ابر مقایسه شده است. کلیه آزمایش‌ها بر روی سیستمی با مشخصات پردازنده IntelCoreI5-3MB و حافظه 8GB DDR 3 انجام شده است.

۴-۱- محیط شبیه‌سازی

برای این که بتوان محیط ابر را با الگوریتم‌های مبتنی بر آن مورد ارزیابی قرار داد، نیاز به آزمایش‌های کنترل‌شده و مکرری است به طوری که انجام آن در محیط‌های واقعی کلود به دلیل خصوصیت ناهمگونی و پویایی آن مشکل است. به علاوه محیط آزمایشگاهی ابر دارای محدودیت‌هایی است و ایجاد یک محیط آزمایشگاهی با اندازه و قابلیت دلخواه از لحاظ زمان و مالی هزینه‌بر است. هم‌چنین این عمل نیاز به کنترل و مدیریت سیاست‌های متنوعی در هر یک از منابع موجود است. به همین دلیل استفاده از شبیه‌سازها برای بررسی ارزیابی و کارآیی الگوریتم‌های پیشنهادی در محیط ابر مورد توجه قرار گرفته است. کلودسیم^۱ یک ابزار شبیه‌سازی در محیط ابر است (Calheiros et al., 2011). این ابزار، یک شبیه‌ساز مبتنی بر جاوا است که ویژگی‌هایی را برای خدمات زمان‌بندی و تخصیص منابع و هم‌چنین واسطی برای تخصیص کارها به منابع در محیط ابر مهیا کرده است. هم‌چنین این ابزار دارای قابلیت مدل‌سازی پیکربندی‌های متنوعی از منابع است. شبکه‌های ارتباطی به عنوان اجزای اصلی ابر به کار می‌روند. در محیط ابر منابع از طریق شبکه‌های ارتباطی و تجاری به یکدیگر متصل هستند. محیط‌های شبیه‌سازی ابر باید قادر به شبیه‌سازی عملیات به اشتراک‌گذاری منابع، سرویس‌دهی و مکانیزم‌های ارسال و دریافت اطلاعات باشند که کلودسیم از این توانایی برخوردار است. نکته‌ای که بسیار حائز اهمیت است این است که وقتی یک ساختار بزرگ کلود در محیط کلودسیم پیاده‌سازی می‌شود سازمان‌دهی و مدیریت گره‌ها و کاربرها و مدیریت منابع در سطح بالا^۲ قرار می‌گیرند و باید در شبیه‌سازی مورد توجه قرار گیرند.

۴-۲- معیارهای استفاده‌شده برای ارزیابی

زمان اجرای کل^۳: به میانگین فاصله زمانی آغاز و پایان کلیه کارها اطلاق می‌شود.
 هزینه اجرا^۴: هزینه‌ای که برای اجرای کامل کارها از نظر مالی به کاربر تحمیل می‌شود و ارتباط تنگاتنگی با معیار *makespan* دارد؛ به گونه‌ای که کاهش *makespan* باعث کاهش این معیار خواهد شد.
 ضریب عدم تعادل^۵: این معیار عدم تعادل و ناهماهنگی بین ماشین‌های مجازی را بیان می‌کند که با رابطه ۳ محاسبه می‌گردد:

$$DI = \frac{T_{max} + T_{min}}{T_{avg}} \quad (3)$$

در این فرمول T_{max} ، T_{min} و T_{avg} به ترتیب بیش‌ترین، کم‌ترین و میانگین زمان اجرای کل در میان ماشین‌های مجازی را بیان می‌کنند.

واحد پارامترهای شبیه‌سازی در جدول ۲ نشان داده شده است.

-
1. CloudSim
 2. Highlevel
 3. Total execution time (Makespan)
 4. Execution cost
 5. Degree of imbalance

جدول (۲): واحد پارامترها

نام پارامتر	واحد
DI	Without any Dimension
Makespan	Time in second
VMM	Mips (Millions Of Instruction Per Second)
VMF	Mips (Millions Of Instruction Per Second)
BW	Bit per second
JL	Mega byte

۴-۳- نتایج شبیه‌سازی در سناریوی اول

در سناریوی اول، بازدهی الگوریتم پیشنهادی با پارامترهای یکسان با دو الگوریتم SGA^1 و MGA^2 مقایسه می‌شود (Kaur & Verma, 2012). در این دو روش از الگوریتم ژنتیک برای زمان‌بندی در محیط ابر استفاده شده است. در الگوریتم SGA در ابتدا یک جمعیت اولیه به صورت تصادفی انتخاب می‌گردد. سپس با استفاده از تابع fitness میزان مناسب بودن ماشین مجازی برای انتساب به کار ارسالی محاسبه می‌گردد. سپس بین کروموزوم‌های انتخاب شده عمل crossover انجام می‌گیرد. در این مرحله یک کروموزوم انتخاب می‌گردد که باید بر روی آن عملگر جهش انجام گیرد. سپس با استفاده از تابع fitness میزان مناسب بودن کروموزوم‌های انتخاب شده محاسبه می‌گردد و بهترین آن‌ها (کروموزوم‌هایی با میزان تابع fitness بالاتر) به جمعیت اولیه اضافه می‌شود و جمعیت جدید را تشکیل می‌دهد. این عملیات تا برقراری شرط پایان الگوریتم ادامه می‌یابد. در الگوریتم MGA که در حقیقت بهبود یافته الگوریتم ژنتیک استاندارد است، در مرحله انتخاب کروموزوم معیار میانگین زمان اجرای کارها در نظر گرفته می‌شود. این الگوریتم، الگوریتم ژنتیک استاندارد را در معیارهای میانگین زمان اجرای کارها و هزینه اجرا بهبود داده است.

۴-۳-۱- آزمایش اول: مقایسه میانگین زمان اجرای کارها

در آزمایش اول، بازدهی الگوریتم پیشنهادی در معیار میانگین زمان اجرای کارها با پارامترهای یکسان با دو الگوریتم ذکر شده مقایسه شده است. هدف اصلی این پژوهش انتساب کارها به ماشین‌های مجازی با در نظر گرفتن طول کار، توان محاسباتی، پهنای باند و هم‌چنین حافظه در ماشین مجازی است و اعمال پارامترهای ذکر شده باعث کاهش میانگین زمان اجرای کارها می‌گردد. در این آزمایش پارامترها، طبق جدول (۳) اعمال شده است.

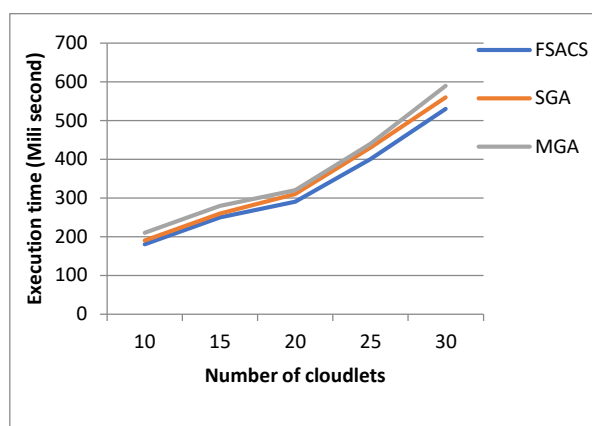
جدول (۳): شرایط پارامترهای آزمایش اول و دوم

حافظه ماشین مجازی	پهنای باند	توان محاسباتی هر گره	تعداد کارها	محدوده طول کارها	تعداد پردازنده‌ها
۲۰۴۸-۵۱۲	۱۰۰۰-۵۰۰	۱۰۰-۵۰۰	۱۰-۳۰	۱۰۰۰۰-۰۰۰۰	۲

نتایج شبیه‌سازی و مقایسه الگوریتم پیشنهادی با الگوریتم‌های SGA و MGA در جدول ۴ و شکل ۱۱ نشان داده شده است. لازم به ذکر است که در عملیات شبیه‌سازی برای هر یک از تعداد کارهای داده شده ۱۰ بار عملیات شبیه‌سازی انجام گرفته و میانگین نتایج آن‌ها در جدول قید گردیده است.

جدول (۴): مقایسه میانگین زمان اجرای کارها در روش پیشنهادی با دیگر روش‌ها

تعداد کار	روش پیشنهادی (FSACS)	SGA	MGA
۱۰	۱۸۰	۲۱۰	۱۹۰
۱۵	۲۵۰	۲۸۰	۲۶۰
۲۰	۲۹۰	۳۲۰	۳۱۰
۲۵	۴۰۰	۴۴۰	۴۳۰
۳۰	۵۳۰	۵۹۰	۵۶۰
میانگین	۳۳۰	۳۶۸	۳۵۰



شکل (۱۱): میانگین زمان اجرای کارها

چنانچه مشاهده می‌شود الگوریتم پیشنهادی میانگین زمان اجرای کارهای بهتری نسبت به دو الگوریتم SGA و MGA دارد.

۴-۳-۲- آزمایش دوم: مقایسه هزینه اجرای کارها

در آزمایش دوم، بازدهی الگوریتم پیشنهادی در معیار هزینه اجرای کارها با پارامترهای یکسان با دو الگوریتم ذکر شده مقایسه شده است. از آنجایی که معیار اصلی طرح پیشنهادی، استفاده بهینه از منابع محاسباتی است. منابع با توان محاسباتی بالا فقط در زمانی استفاده می‌شوند که کار با نیاز پردازشی بالا مورد نیاز باشد. در حقیقت طرح پیشنهادی کارهای با طول محاسباتی بالا را به ماشین‌های مجازی با توان پردازشی بالا و کارهای با نیاز محاسباتی کم را به منابع با توان پردازشی کم انتساب می‌دهد. جدول ۵ هزینه منابع با توان محاسباتی مختلف را نشان می‌دهد.

جدول (۵): لیست هزینه منابع محاسباتی

۵۰۰	۴۰۰	۳۰۰	۲۰۰	۱۰۰	Processor Capacity(Mips)
۴۰	۳۰	۲۵	۲۰	۱۵	Per Unit Cost(\$)

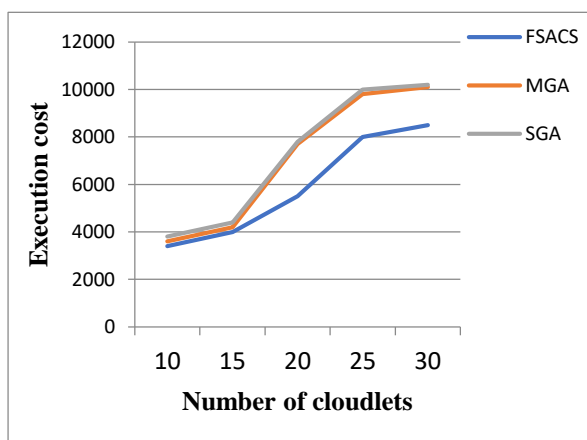
منابع (ماشین‌های مجازی) موجود در محیط ابر دارای شرایط مختلفی هستند، از جمله این شرایط می‌توان به هزینه‌های مختلف در واحد زمان و نیز قدرت پردازشی متفاوت اشاره کرد (Avram, 2014). هدف اصلی طرح این است که با در نظر گرفتن منابعی که تعداد کارهای کمتری اجرا می‌کنند مانع از به هدر رفتن توان محاسباتی آزاد سیستم، بار سیستم و هزینه ارائه‌کننده برای ارائه یک سرویس متعادل گردند. در حالت کلی هزینه ارائه‌کننده برای

ارائه یک سرویس شامل هزینه خرید زیرساخت‌ها، هزینه ایجاد سرویس، هزینه نگهداری و پشتیبانی و غیره می‌باشد و در نظر گرفتن این پارامتر این هدف را دنبال می‌کند که با کمتر شدن این هزینه، مقدار سود ارائه‌کننده افزایش پیدا کند؛ اما در این مقاله، برای سادگی کار از پارامتر هزینه سخت‌افزار (پردازنده) برای تعیین هزینه اجرا استفاده شده است. در نظر گرفتن توان محاسباتی منابع و طول کار دریافتی باعث بهبود توازن بار سیستم خواهد شد. هم‌چنین در نظر گرفتن هزینه ارائه‌کننده برای ارائه یک سرویس به عنوان ورودی سیستم فازی و در نظر گرفتن این پارامتر به همراه پارامترهای ذکر شده باعث رسیدن به نقطه تعادل هزینه ارائه‌کننده برای ارائه یک سرویس می‌گردد.

نتایج شبیه‌سازی و مقایسه الگوریتم پیشنهادی با الگوریتم‌های SGA و MGA در جدول ۶ و شکل ۱۲ نشان داده شده است. لازم به ذکر است که در عملیات شبیه‌سازی برای هر یک از تعداد کارهای داده شده ۱۰ بار عملیات شبیه‌سازی انجام گرفته و میانگین نتایج آن‌ها در جدول قید گردیده است.

جدول (۶): مقایسه هزینه اجرای کارها در الگوریتم‌های SGA و MGA و FSACS

تعداد کار	روشن پیشنهادی (FSACS)	SGA	MGA
۱۰	۳۴۰۰	۳۸۰۰	۳۶۰۰
۱۵	۴۰۰۰	۴۴۰۰	۴۲۰۰
۲۰	۵۵۰۰	۷۸۰۰	۷۷۰۰
۲۵	۸۰۰۰	۱۰۰۰۰	۹۸۰۰
۳۰	۸۵۰۰	۱۰۲۰۰	۱۰۱۰۰
میانگین	۶۷۸۰	۷۲۴۰	۷۰۸۰



شکل (۱۲): هزینه اجرای کارها

۴-۴- نتایج شبیه‌سازی در سناریوی دوم

در سناریوی دوم، بازدهی الگوریتم پیشنهادی با پارامترهای یکسان با الگوریتم ذوب فلزات مقایسه شده است (Moschakis & Karatza, 2015). طبق الگوریتم ذوب فلزات، هر فضای جستجو می‌تواند جواب مسئله باشد. وضعیت اولیه از یک وضعیت به وضعیت همسایه در فضای جستجو تغییر می‌کند. در این الگوریتم به وسیله یک تابع وضعیت همسایه با وضعیت فعلی مقایسه می‌شود و بازدهی الگوریتم بستگی زیادی به این تابع دارد. چنانچه وضعیت همسایه بهتر از وضعیت فعلی باشد وضعیت همسایه به عنوان جواب در نظر گرفته می‌شود. به زبان ساده‌تر منبع و ماشین مجازی مجاور برای کار وارد شده در نظر گرفته می‌شود. برای در نظر گرفتن مجدد وضعیت همسایه چنانچه وضعیت

همسایه از وضعیت فعلی بهتر نباشد پارامترهای کیفیت خدمات را در نظر می‌گیریم؛ طوری که مثلاً اگر دسترس‌پذیری وضعیت همسایه که یکی از پارامترهای کیفیت خدمات است، از وضعیت فعلی بهتر باشد، منبع وضعیت همسایه به‌عنوان جواب برای کار وارد شده در نظر گرفته می‌شود.

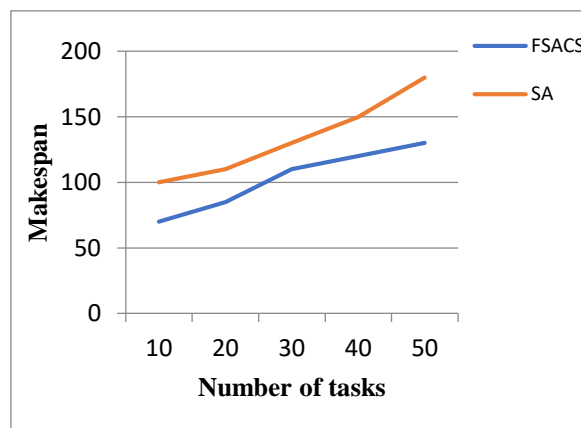
در این سناریو برای مقایسه دو وضعیت ذکر شده از تابع فازی استفاده می‌شود. در الگوریتم ذوب فلزات در نظر گرفته شده که برای مقایسه استفاده می‌شود بدون در نظر گرفتن طول کار وارد شده، چنانچه توان محاسباتی ماشین مجازی همسایه بهتر از ماشین مجازی فعلی باشد ماشین مجازی همسایه به‌عنوان جواب در نظر گرفته می‌شود.

۴-۴-۱- آزمایش اول: مقایسه میانگین زمان اجرای کارها

نتایج شبیه‌سازی و مقایسه الگوریتم پیشنهادی با الگوریتم ذوب فلزات در جدول ۷ و شکل ۱۳ نشان داده شده است. لازم به ذکر است که در عملیات شبیه‌سازی برای هریک از تعداد کارهای داده شده ۱۰ بار عملیات شبیه‌سازی انجام گرفته و میانگین نتایج آن‌ها در جدول قید گردیده است.

جدول (۷): مقایسه میانگین زمان اجرای کارها در طرح پیشنهادی و الگوریتم ذوب فلزات

تعداد کارها	روش پیشنهادی (FSACS)	Simulated annealing
۱۰	۷۰	۱۰۰
۲۰	۸۵	۱۱۰
۳۰	۱۱۰	۱۳۰
۴۰	۱۲۰	۱۵۰
۵۰	۱۳۰	۱۸۰
میانگین	۱۰۳	۱۳۴



شکل (۱۳): میانگین زمان اجرای کارها

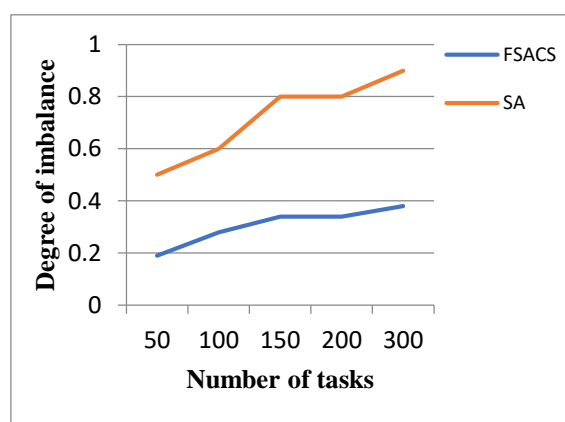
۴-۴-۲- آزمایش دوم، مقایسه ضریب عدم تعادل

در این آزمایش، بازدهی الگوریتم پیشنهادی در معیار ضریب عدم تعادل با پارامترهای یکسان با الگوریتم فلزات ذوب شده مقایسه شده است. از آنجایی که معیار مقدار مصرف شده پردازنده و طول کار دریافتی، دو معیار اصلی ارزیابی هستند؛ فاصله بین حداقل زمان اجرای کارها و حداکثر اجرای کارها و هم‌چنین میانگین آن‌ها بسیار کم خواهد بود و در نتیجه این معیار از دو روش قبلی مقدار کمتری را در بر خواهد گرفت. جدول ۸ و شکل ۱۴ این موضوع را نشان

می‌دهند. لازم به ذکر است که در عملیات شبیه‌سازی برای هریک از تعداد کارهای داده‌شده ۱۰ بار شبیه‌سازی انجام گرفته و میانگین نتایج آن‌ها در جدول قید گردیده است.

جدول (۸): مقایسه ضریب عدم تعادل در طرح پیشنهادی و الگوریتم ذوب فلزات

تعداد کار	روش پیشنهادی (FSACS)	Simulated annealing
۵۰	۰/۱۹	۰/۵
۱۰۰	۰/۲۸	۰/۶
۱۵۰	۰/۳۴	۰/۸
۲۰۰	۰/۳۴	۰/۸
۳۰۰	۰/۳۸	۰/۹
میانگین	۰/۳۰	۰/۷۲



شکل (۱۴): ضریب عدم تعادل

۴-۵- نتایج شبیه‌سازی در سناریوی سوم

در این آزمایش روش پیشنهادی با fuzzy neural network در شرایط یکسان مقایسه شده است (Kumar & Dinesh, 2012). پارامترهای ورودی در روش fuzzy neural network، پهنای باند، مقدار حافظه و هم‌چنین مقدار مصرفی پردازنده می‌باشد که در معیار میانگین زمان اجرای کارها بهبود حاصل شده است. طرح و روش پیشنهادی شباهت بسیاری با این طرح دارد و نتایجی که به دست آمده است نزدیک به یکدیگر می‌باشند. در این آزمایش پارامترها، طبق جدول ۹ اعمال شده است.

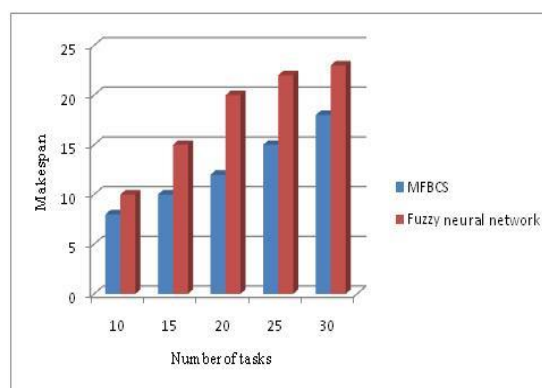
جدول (۹): شرایط پارامترهای آزمایش پنجم

تعداد ماشین مجازی	پهنای باند	توان محاسباتی هر گره	تعداد کارها	محدوده طول کارها	تعداد پردازنده‌ها
۱۰-۳۵	۵۰۰-۱۰۰۰	۵۰۰-۲۰۰۰	۱۰-۳۵	۱۰۰۰-۲۰۰۰۰	۵

نتایج شبیه‌سازی و مقایسه الگوریتم پیشنهادی با الگوریتم fuzzy neural network در جدول ۱۰ و شکل ۱۵ نشان داده شده است. لازم به ذکر است که در عملیات شبیه‌سازی برای هریک از تعداد کارهای داده‌شده ۱۰ بار عملیات شبیه‌سازی انجام گرفته و میانگین نتایج آن‌ها در جدول قید گردیده است.

جدول (۱۰): مقایسه میانگین زمان اجرای کارها در الگوریتم‌های MFBCS و fuzzy neural network

تعداد کار	روش پیشنهادی (FSACS)	Fuzzy neural network
۱۰	۸	۱۰
۱۵	۱۰	۱۵
۲۰	۱۲	۲۰
۲۵	۱۵	۲۲
۳۰	۱۸	۲۳
میانگین	۱۲/۶	۱۸



شکل (۱۵): میانگین زمان اجرای کارها

۵- نتیجه‌گیری و پیشنهاداتی آتی

در محیط ابر انواع و اقسام رایانه‌های مختلف با توانایی‌ها و قابلیت‌های مختلف و سیستم‌عامل‌های متفاوت یافت می‌شوند. سیستم‌های توزیع‌شده نظیر ابر دارای گره‌های محاسباتی خودمختار و ناهمگن هستند و دسترس‌پذیری آن‌ها در هر زمان تغییر می‌کند. پویایی این محیط، کار را مشکل ساخته است؛ بنابراین باید به دنبال ارائه راه‌کاری برای زمان‌بندی در این محیط ناهمگن باشیم. در این مقاله یک الگوریتم زمان‌بندی ترکیبی با کمک سیستم فازی و روش ذوب فلزات ارائه شد که با در نظر گرفتن هم‌زمان پارامترهای پهنای باند، میزان توان محاسباتی، میزان تأخیر و هم‌چنین طول کار درخواستی با حفظ توازن بار سعی در کاهش زمان خاتمه کارها دارد. هدف الگوریتم پیشنهادی رسیدن به یک چارچوب زمان‌بندی است به صورتی که زمان خاتمه کارهای مناسبی داشته باشد.

نتایج شبیه‌سازی نشان می‌دهد که الگوریتم پیشنهادی، زمان خاتمه کارهای کمتری نسبت به الگوریتم‌های مشابه دارد. کاهش هزینه مصرف‌شده و هم‌چنین کاهش میزان ضریب عدم تعادل از ویژگی‌های دیگر طرح پیشنهادی است. در زمینه زمان‌بندی منبع، کارهای جدیدتری نیز می‌توان انجام داد. به‌خصوص در مورد تعادل بار که یکی از عوامل کیفیت خدمات می‌باشد. با این مطالعات می‌توان به نتایج درخشان‌تری امیدوار بود. در مورد کارهای آتی و ادامه تحقیقات در این زمینه موارد زیر پیشنهاد می‌شود:

- در مورد ایده پیشنهادشده این مقاله می‌توان معیارهای دیگری از پارامترهای تعادل بار را که منجر به بهبود کیفیت خدمات می‌شود؛ اعمال کرد. به عنوان مثال زمان انتظار کارها در صف آماده کامپیوترهای میزبان دوردست.
- با استفاده از روش‌های مبتنی بر الگوریتم‌های ژنتیک در آینده می‌توان بهبودهای قابل توجهی در الگوریتم پیشنهادی ایجاد کرد؛ زیرا این روش‌ها در بهینه‌سازی مسائل کمک بسیاری می‌کنند.

منابع

اکبری. محمد کاظم و سرگزایی جوان، مرتضی. (۱۳۸۹). محاسبات ابری ارائه معماری‌ها، ابزارها، سرویس‌ها و مسائل مرتبط. دانشگاه صنعتی امیرکبیر.

- Avram, M. G. (2014). Advantages and challenges of adopting cloud computing from an enterprise perspective. *Procedia Technology*, 12, 529-534.
- Aymerich, F. M., Fenu, G., & Surcis, S. (2008, August). An approach to a cloud computing network. In *2008 First International Conference on the Applications of Digital Information and Web Technologies (ICADIWT)* (pp. 113-118). IEEE.
- Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., & Buyya, R. (2011). CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and experience*, 41(1), 23-50.
- Chen, M., Li, M., & Cai, F. (2010, August). A model of scheduling optimizing for cloud computing resource services based on Buffer-pool Agent. In *2010 IEEE International Conference on Granular Computing* (pp. 107-110). IEEE.
- Jin, J., Luo, J., Song, A., Dong, F., & Xiong, R. (2011, May). Bar: An efficient data locality driven task scheduling algorithm for cloud computing. In *2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing* (pp. 295-304). IEEE.
- Kaur, S., & Verma, A. (2012). An efficient approach to genetic algorithm for task scheduling in cloud computing environment. *International Journal of Information Technology and Computer Science (IJITCS)*, 4(10), 74-79.
- Kumar, V. V., & Dinesh, K. (2012). Job scheduling using fuzzy neural network algorithm in cloud environment. *Bonfring International Journal of Man Machine Interface*, 2(1), 1-6.
- LD, D. B., & Krishna, P. V. (2013). Honey bee behavior inspired load balancing of tasks in cloud computing environments. *Applied soft computing*, 13(5), 2292-2303.
- Li, Y., Zhang, H., & Kim, K. H. (2011, December). A power-aware scheduling of mapreduce applications in the cloud. In *2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing* (pp. 613-620). IEEE.
- Lin, G., Fu, D., Zhu, J., & Dasmalchi, G. (2009). Cloud computing: IT as a service. *IT professional*, 11(2), 10-13.
- Liu, S., Gao, P., Mu, K., & Lu, Y. (2014, January). The Deployment of Resource Pool Based on Hyper-V Virtualization in Cloud Computing. In *2014 International Conference on Computer, Communications and Information Technology (CCIT 2014)* (pp. 69-72). Atlantis Press.
- Moschakis, I. A., & Karatza, H. D. (2015). Towards scheduling for Internet-of-Things applications on clouds: a simulated annealing approach. *Concurrency and Computation: Practice and Experience*, 27(8), 1886-1899.
- Paul, M., & Sanyal, G. (2011, December). Survey and analysis of optimal scheduling strategies in cloud environment. In *2011 World Congress on Information and Communication Technologies* (pp. 789-792). IEEE.
- Sugeno, M. (1985). An introductory survey of fuzzy control. *Information sciences*, 36(1-2), 59-83.
- Wu, H., Tang, Z., & Li, R. (2012, February). A priority constrained scheduling strategy of multiple workflows for cloud computing. In *2012 14th International Conference on Advanced Communication Technology (ICACT)* (pp. 1086-1089). IEEE.

