APJ. 4(4):30-45, Winter 2023

Arman Process Journal (APJ)

Homepage: armanprocessjournal.ir

A Review of Load Balancing Algorithms in Cloud Computing Environment

WahabAminiazar[1*], RasoulFarahi[2], Fatmeh Dashti[3] and Kamal Rahmani[4]

[1] Assistant Professor, Department of Computer Engineering and Information Technology, Faculty of Electrical and Computer Engineering , Islamic Azad University, Mahabad Branch, Mahabad, Iran

[2] Instructor, Department of Computer Engineering and Information Technology, Faculty of Electrical and Computer Engineering, Islamic Azad University, Mahabad Branch, Mahabad, Iran.

[3] Tabriz Electricity Distribution Company, Tabriz, Iran

[4] Assistant Professor, Department of Civil Engineering, Faculty of Civil Engineering, Islamic Azad University, Mahabad Branch, Mahabad, Iran

Received: 17 December 2023
Accepted: 22 March 2024

KEYWORDS:
Cloud Computing,
Virtualization,
Task Scheduling,
Load Balancing,
Resource Optimization.

[1] Corresponding author
Aminiazar@iau-mahabad.ac.ir

ABSTRACT

The instantaneous increase in users and their need for internet services caused that, in a short time, the companies that provided this type of service faced problems such as the inability to respond quickly to users and the increase in their costs. Therefore, many of these companies, with a lot of investments in research fields, thought of effective and cost-effective ways to serve a high volume of users, and in this way, new technology and an efficiency system called cloud computing were created. With the increase in users using cloud computing services and therefore the increase in the number of requests, in order to achieve the mentioned benefits, there is a need to establish appropriate mechanisms for load balancing, work scheduling and virtualization. Sazi is in cloud computing. This load can include memory capacity, network load or delay. Load balancing is the process of distributing load among different nodes of a distributed system in order to improve the utilization of resources and response time, while it is a situation in which some nodes have a heavy load while the node Others avoid being unemployed or having very little work to do. Considering the necessity and importance of load balancing in cloud computing, in this article, a comprehensive review of static, dynamic and nature-inspired algorithms for load balancing in a cloud space to handle the response time of data centers and their overall performance is given. We pay by analyzing the load balancing algorithms. We show that the ant colony algorithm, the genetic algorithm and the particle swarm optimization algorithm with optimal allocation of tasks can play a more effective role in balancing the load in the cloud space. Also, the results show that CloudSim software has been used the most in simulating load balancing algorithms in the cloud space.

NUMBER OF REFERENCES	NUMBER OF FIGURES	NUMBER OF TABLES
44	1	1

نشریه تخصصی آرمان پردازش، دوره ۴، شماره ۴، زمستان ۱۴۰۲

فصلنامه تخصصی آرمان پردازش
(APJ)
Homepage: www.armanprocessjournal.ir

مروری بر الگوریتم‌های متعادل کردن بار در محیط رایانش ابری

وهاب امینی‌آذر[۱*]، رسول فرحی[۲] ، فاطمه دشتی[۳] و کمال رحمانی

[۱*] استادیار ، گروه مهندسی کامپیوتر و فناوری اطلاعات، دانشکده مهندسی برق و کامپیوتر، دانشگاه آزاد اسلامی واحد مهاباد، مهاباد، ایران

[۲] مربی، گروه مهندسی کامپیوتر و فناوری اطلاعات، دانشکده مهندسی برق و کامپیوتر، دانشگاه آزاد اسلامی واحد مهاباد، مهاباد، ایران

[۳] شرکت توزیع نیروی برق تبریز، تبریز، ایران

[۴] استادیار، گروه مهندسی عمران، دانشکده مهندسی عمران، دانشگاه آزاد اسلامی، واحد مهاباد، مهاباد، ایران

چکیده

افزایش لحظه‌ای کاربران و نیاز آن‌ها به خدمات اینترنتی باعث شد که در اندک زمانی شرکت‌هایی که این گونه از خدمات را ارائه می‌دادند با مشکلاتی نظیر عدم توانایی در پاسخگویی سریع به کاربران و افزایش هزینه‌هایشان روبرو شوند. از این‌رو بسیاری از این شرکت‌ها با سرمایه‌گذاری‌های فراوان در زمینه‌های تحقیقاتی به فکر شیوه‌های موثر و مقرون به صرفه، برای سرویس‌دهی به حجم بالایی از کاربران افتادند و به این ترتیب فناوری نوین و کارآمدی به نام رایانش ابری به وجود آمد. با افزایش کاربران استفاده کننده از خدمات رایانش ابری و لذا افزایش میزان درخواست‌ها، جهت دستیابی به مزایای اشاره شده، نیاز به استقرار مکانیزم‌های مناسبی جهت توازن بار، زمانبندی کار و مجازی‌سازی در رایانش ابری می‌باشد. این بار می‌تواند شامل ظرفیت حافظه، بار شبکه و یا تاخیر باشد. توازن بار فرایند توزیع بار در میان گره‌های مختلف یک سیستم توزیع شده جهت بهبود بهره‌برداری از منابع و زمان پاسخ می‌باشد در حالی که از وضعیتی که درآن برخی از گره‌ها دارای بار سنگین باشند در حالی‌که گره‌های دیگر بیکار باشند و یا کار خیلی کمی برای انجام دادن داشته باشند اجتناب می‌کند. با توجه به ضرورت و اهمیت توازن بار در رایانش ابری، در این مقاله به مروری جامع از الگوریتم‌های استاتیک، پویا و الهام‌گرفته از طبیعت جهت متعادل‌سازی بار در یک فضای ابری برای رسیدگی به زمان پاسخ‌دهی مراکز داده و عملکرد کلی آن‌ها می‌پردازیم. با تحلیل الگوریتم‌های تعادل بار نشان می‌دهیم که الگوریتم کلونی مورچه ، الگوریتم ژنتیک و الگوریتم بهینه‌سازی ازدحام ذرات با تخصیص بهینه وظایف می‌توانند نقش مؤثرتری در متعادل‌سازی بار در فضای ابری را داشته باشند. همچنین نتایج گویا این است که نرم‌افزار کلودسیم بیشترین کاربرد را در شبیه‌سازی الگوریتم‌های متعادل کردن بار در فضای ابری داشته است.

واژگان کلیدی:
رایانش‌ابری،
مجازی‌سازی،
زمان‌بندی وظایف،
تعادل بار،
بهینه‌سازی منابع.

تعداد جداول	تعداد شکل ها	تعداد مراجع
۱	۱	۴۴

مقدمه

رایانش ابری[۱] علیرغم مزایای فراوانی که دارد برای تحقق یافتن و پیاده-سازی با چالش‌های زیادی مواجه است. در میان این چالش‌ها توازن بار یکی از مسائل مهم در رایانش ابری است که نقش بسزایی در کارایی، بهره‌وری از منابع و زمان پاسخ کارها دارد بهبود کارایی سیستم یکی از مسائل مهم در سیستم‌های رایانش ابری توزیع‌شده می‌باشد. این امر همواره نیازمند به اشتراک‌گذاری بارکاری در میان گره‌های محاسباتی مختلف سیستم توزیع شده جهت استفاده موثر از منابع قابل دسترس می‌باشد. توزیع تصادفی کارهای ورودی به گره‌های محاسباتی در سیستم توزیع شده ممکن است منجر به وضعیتی شود که در آن برخی از گره‌ها دارای بار سنگین شوند در حالی که سایر گره‌ها دارای بار سبک و یا حتی بیکار باشند. بنابراین در چنین وضعیتی انتقال برخی از کارها از گره‌های با بار سنگین به گره‌های با بار سبک و یا بیکار جهت استفاده موثر از منابع محاسباتی قابل دسترس در محیط سیستم‌های محاسباتی توزیع شده، مطلوب بوده و تاثیر مهمی بر کارایی دارد به عبارت دیگر، توازن بار فرآیند اختصاص و تخصیص مجدد بار در میان منابع قابل دسترس، با هدف بهبود بهره‌وری از منابع[۲] و در نتیجه کارایی می‌باشد. توافقنامه سطح خدمات و رضایت‌مندی کاربران می‌تواند به وسیله تکنیک‌های توازن بار[۳] مناسب محقق گردد. بنابراین ارائه الگوریتم‌ها و مکانیزم‌های توازن بار مناسب یک عامل مهم جهت موفقیت محیط‌های رایانش ابری می‌باشد. با توجه به اهمیت این موضوع در این مقاله به بیان اهداف، انواع روش‌های توازن بار، سیاست‌های موجود در آن و نیز دسته‌بندی الگوریتم‌های توازن بار و مقایسهٔ آن‌ها خواهیم پرداخت تا بتوانیم با بررسی بر روی آن‌ها، زمینه را برای ارائه الگوریتم‌های توازن بار کاراتر در آینده فراهم کنیم. سازمان‌دهی مقاله به صورت زیر است: در بخش دوم مروری بر مفاهیم و مقدمات رایانش ابری خواهیم داشت و در ادامه و در بخش سوم مفاهیم، مزایای ، چالش-ها و معیارهای توازن بار مورد بحث قرار می‌گیرد. در بخش چهارم انواع الگوریتم‌های توازن بار و دسته‌بندی آن‌ها بیان شده است. در بخش پنجم پیشینه تحقیق و در نهایت در بخش ششم نتیجه‌گیری و کارهای آینده ارائه شده است.

رایانش ابری

رایانش ابری یک الگوی تکاملی است که بسیاری از محققان در تلاش هستند تا تعریفی جامع را برای آن ارائه دهند. برخی بر این باورند که فقط یک نام جدید به سرویس داده شده است. موسسه ملی فناوری و استانداردها[۴] رایانش ابری را اینگونه تعریف می‌کند: " رایانش ابری مدلی است برای فراهم کردن دسترسی آسان و براساس تقاضای کاربر، از طریق شبکه به مجموعه‌ای از منابع رایانشی قابل تغییر و پیکربندی است که این دسترسی می‌تواند با کمترین نیاز به مدیریت منابع و یا نیاز به دخالت مستقیم به سرعت فراهم گردد [۱].

رایانش ابری می‌تواند برای تحویل سرویس‌های موجود در لایه‌های مختلف از سخت‌افزار گرفته تا برنامه‌های کاربردی مورد استفاده قرار گیرد. این مدل‌ها بسته به نیاز مشتریان می‌باشد و یک مشتری می‌تواند یک یا چند سرویس را به صورت همزمان استفاده کند. طبق تعریف ارائه شده توسط NIST سرویس‌های مختلف ابر را می‌توان در سه گروه اصلی تقسیم‌بندی کرد که عبارتند از [۲]:

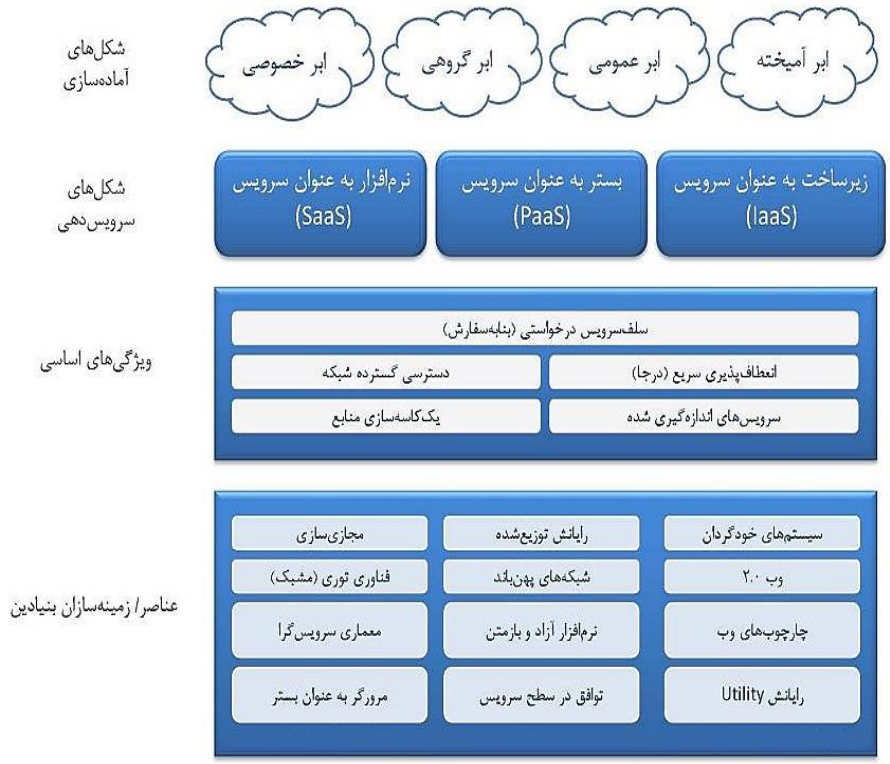
- **زیرساخت به عنوان سرویس[۴]:** فروشندگان ابر منابعی مثل فضای ذخیره‌سازی، سخت افزار، سرور و اجرای شبکه را در اختیار مشتری قرار می‌دهند. نگهداری این منابع سخت افزاری بر عهده فروشنده ابر است مدل زیرساخت به عنوان سرویس، به صورت پویا تخصیص پهنای باند و منابع سرور را برای ابر انجام می دهد. این سرویس همچنین به ابر اجازه می‌دهد تا در مدت زمانی که ترافیک بالاست، منابع را به صورت پویا و بر اساس شرایط خواسته شده افزایش دهد. ابر محاسباتی با قابلیت توسعه آمازون (EC2) و Secure Storage Service (53) مثال‌هایی از ارائه زیر ساخت به عنوان سرویس هستند [۲].
- **پلت فرم به عنوان سرویس[۵]:** شامل سرویس هایی برای توسعه، تست، پیاده سازی، میزبانی و مدیریت برنامه های کاربردی جهت پشتیبانی از برنامه ها در طول چرخه حیات است. مدل پلت فرم به عنوان سرویس تمام ساختارهای لازم برای اجرای برنامه های کاربردی در اینترنت را فراهم می کند. Google App Engine و Microsoft Azure مثال هایی از پلت فرم به عنوان سرویس هستند [۲].
- **نرم افزار به عنوان سرویس[۶]:** یک برنامه کاربردی را به کاربر پایانی ارائه می دهد. کاربر پایانی نیازی به درک زیرساخت ها ندارد و به سادگی از برنامه استفاده می کند. تمام جزئیات مربوط به برنامه تا زیرساخت به عنوان یک سرویس در پشت صحنه قرار می گیرد. Salesforce.com، Oracle CRM On Demand و Netsuit مثال هایی از ارائه دهندگان نرم افزار به عنوان سرویس هستند [۳].

از طرفی ابرها می‌توانند براساس مدل توسعه و استقرار به چهار دسته طبقه‌بندی شوند و مدل‌های مختلف ابر از لحاظ معماری و مکان دیتاسنتر و نوع مالکیت باهم متفاوت هستند. انواع مدل‌های توسعه ابر عبارتند از : ابر عمومی، ابر خصوصی، ابر ترکیبی، ابر اجتماعی [۴].

- **ابر خصوصی[۷]:** ابر اختصاصی زیرساخت های این ابر برای استفاده انحصاری توسط یک سازمان واحد متشکل از چندین مصرف کننده ساخته شده است. این مدل ابر به سازمانها اجازه می دهد، منابعشان را بر روی شبکه خصوصی خود مدیریت کنند. شرکت

یا سازمان خود مالک سرویس است و همچنین سازمان است که اجازه دسترسی کاربران به سرویس را می دهد [۴].

- **ابر عمومی**[۸]: زیر ساخت های این ابر برای استفاده عمومی فراهم شده است. این مدل ابر ممکن است متعلق به یک تجارت، سازمان دولتی، دانشگاه و یا ترکیبی از آنها باشد و مدیریت آن توسط مالک ابر انجام می شود. در این مدل، ارائه دهنده منابعی را که مشتریان نیاز به دسترسی به آن دارند، بر روی یک شبکه عمومی مثل اینترنت قرار می دهد. ارائه دهنده، مالک و اجرا کننده تکنولوژی جهت ارائه سرویس و خدمات است و مشتری هیچ کنترلی بر عملیات سرویس ندارد [۴].
- **ابر ترکیبی**[۹]: این زیرساخت ابری، ترکیبی کامل از زیرساخت‌های ابرهای گوناگون است و به کمک رابط کاربرها با این ابرها تعامل دارد؛ بنابراین سودبردن از مزایای انواع مدل های پیاده سازی را تسهیل می کند؛ بنابراین ابر، حجم کار را بدون هیچ دردسری برای کاربران، میان میزبان خصوصی و عمومی انتقال می دهد [۴].
- **ابر جتماعی**[۱۰] : این ابر برای تعدادی از شرکت‌های ویژه است که منافع، خط مشی ها و منافع انطباقی مشابه را با یکدیگر به اشتراک می‌گذارند. این ابر را یک شرکت یا گروهی از شرکت‌ها مدیریت می‌کنند و در مقایسه با ابر خصوصی، هزینه کمتری دارد، زیرا هزینه میان گروهی از شرکت ها تقسیم می‌شود. [۴].



شکل ۱: انواع ابر و سرویس‌های قابل ارائه در آن [1]

توازن‌بار در رایانش ابری

توازن بار فرآیند توزیع بارهای کاری در بین منابع محاسباتی در یک محیط رایانش ابری و متعادل کردن دقیق ترافیک شبکه برای دسترسی به آن منابع است. متعادل‌سازی بار سازمان‌ها را قادر می‌سازد تا با مسیریابی ترافیک ورودی به چندین سرور، شبکه یا منابع دیگر، نیازهای بار کاری را برآورده کنند، در حالی که عملکرد را بهبود می‌بخشد و از اختلالات در خدمات محافظت می‌کند. تعادل بار همچنین توزیع بارهای کاری را در دو یا چند منطقه جغرافیایی امکان پذیر می کند. توازن بار ابری به شرکت‌ها کمک می‌کند تا به سطوح عملکرد بالا برای هزینه‌های بالقوه کمتر نسبت به فناوری متعادل‌سازی بار سنتی در محل دست یابند. توازن بار ابری از مقیاس پذیری و چابکی ابر برای برآورده کردن نیازهای بارهای کاری توزیع شده با تعداد زیاد اتصالات مشتری بهره می برد. همچنین دسترسی کلی را بهبود می بخشد، توان عملیاتی را افزایش می دهد و تأخیر را کاهش می دهد.علاوه بر حجم کار و توزیع ترافیک، سرویس‌های متعادل کننده بار ابری معمولاً ویژگی‌های دیگری مانند

بررسی سلامت برنامه‌ها، مقیاس‌بندی خودکار و خرابی و مدیریت گواهی یکپارچه ارائه می‌دهند [۶].

۳-۱- مزایای توازن بار در رایانش ابری

در این بخش مزایای و چالش‌های توازن بار در رایانش ابری مورد بحث قرار خواهد گرفت. به طور خلاصه این مزایای شامل موارد زیر است:

- کاهش زمان انتظار کارها
- به حداقل رساندن زمان پاسخ کارها
- به حداکثر رساندن بهرهوری از منابع سیستم
- به حداکثر رساندن توانعملیاتی سیستم
- بهبود قابلیت اطمینان و مقیاس‌پذیری سیستم
- سازگاری با تغییرات آینده [۶].

۳-۲- چالش‌های توازن بار در رایانش ابری

توازن بار در رایانش ابری با چندین چالش روبرو است .در این بخش چالش‌های توازن بار را با هدف طراحی استراتژی‌های توازن بار مناسب در آینده مورد بررسی قرار گرفته است:

۳-۲-۱- مهاجرت ماشین مجازی

طبیعت خدمات برحسب تقاضای محاسبات ابری، دلالت بر این موضوع دارد که هنگامی که یک سرویس وجود دارد منابع باید فراهم گردند . گاهی منابع باید از یک سرور فیزیکی به سرور فیزیکی دیگر که ممکن است در مکانی دور واقع باشد مهاجرت کنند. طراحان الگوریتم های توازن بار، باید به دو موضوع در این رابطه توجه کنند :زمان مهاجرت که بر کارایی تاثیر می گذارد و احتمال حملات [۶].

۳-۲-۲- توزیع فضایی گره‌ها در ابر

گره‌ها در ابر از نظر جغرافیایی توزیع شده می‌باشند. چالش موجود در این رابطه این است که الگوریتم‌های توازن بار باید به گونه‌ای طراحی شوند که پارامترهایی مانند پهنای‌باند شبکه، سرعت‌های ارتباطات، فاصله‌های میان گره‌ها و فاصله بین کاربر و منابع را مورد توجه قرار دهند [۶].

۳-۲-۳- وجود تک نقطه خرابی

برخی از الگوریتم‌های توازن بار متمرکز می‌باشند. در این موارد اگر گره‌ای که الگوریتم را اجرا می‌کند کنترل کننده خراب شود، کل سیستم از کار خواهد افتاد. به خاطر همین تک نقطه خرابی، چالش موجود در اینجا طراحی الگوریتم‌های غیرمتمرکز می‌باشد [۶].

۳-۲-۴- پیچیدگی الگوریتم

الگوریتم‌های توازن بار باید از جنبه‌ای پیاده‌سازی و عملیات ساده باشند. الگوریتم‌های پیچیده دارای تاثیرات منفی در کارایی و عملکرد کلی می‌باشند [۶].

۳-۲-۵- پیداش مراکز کوچک در رایانش ابری

مراکز داده کوچک ارزان‌تر هستند و انرژی کمتری در مقایسهٔ با مراکز داده بزرگ مصرف می‌کنند. بنابراین منابع محاسباتی در سراسر جهان توزیع می‌شوند. چالش موجود در اینجا طراحی الگوریتم‌های توازن بار برای یک زمان پاسخ قابل قبول است [۷].

۳-۲-۶- مدیریت انرژی

بهره‌برداری کم از منابع، هزینهٔ مراکز ابری و توان مصرفی را افزایش می‌دهد. الگوریتم‌های توازن بار باید طوری طراحی شوند که میزان انرژی مصرفی را به حداقل برسانند. بنابراین آنها باید از متدولوژی زمانبندی کار آگاه به انرژی تبعیت کنند. به دلیل تمایل سازمان‌ها و کاربران به استفاده از خدمات ابر، تاسیسات ارائه‌کنندگان ابر در آینده گسترش یافته و از این رو انرژی مصرفی در این صنعت به سرعت افزایش پیدا خواهد کرد. این افزایش در انرژی مصرفی، نه تنها هزینهٔ انرژی را افزایش می‌دهد بلکه انتشار کربن را نیز افزایش می‌دهد. بنابراین مصرف انرژی زیاد تبدیل به یک نگرانی بزرگ برای حوزه صنایع و اجتماع شده است [۷].

۳-۳- معیارهای توازن بار در رایانش ابری

در این بخش، معیارهایی که اکثر نویسندگان در بررسی توازن بار در محیط ابر در نظر می گیرند، ارائه می شود. این معیارها در طراحی و توسعه یک الگوریتم تعادل بار ضروری هستند. این معیارها کیفیت الگوریتم را از نظر عملکرد در برنامه های ابری تعیین می کند [۹].

گذردهی: این معیار جهت محاسبه تعداد فرآیندهای تکمیل شده در واحد زمان می‌باشد.

- **زمان پاسخگویی:** این معیار کل زمانی را که سیستم جهت ارائه خدمت به یک کار تحویل شده صرف می‌کند را اندازه گیری می‌کند.
- **زمان کل:** این معیار جهت محاسبه حداکثر زمان تکمیل کارها یا میزان زمانی که منابع به یک کاربراختصاص یافته‌اند بکار می‌رود. به عبارت دیگر اختلاف بین زمان تکمیل آخرین کار و زمان آغاز اولین کار.
- **مقیاس‌پذیری:** توانایی یک الگوریتم جهت اجرای یکنواخت توازن بار در سیستم بر اساس نیازمندی‌ها با افزایش تعداد گره‌ها است.

- **تحمل‌پذیری خطا:** این معیار قابلیت الگوریتم جهت توازن بار در هنگام وقوع برخی خرابی‌ها در برخی گره‌ها یا لینک‌ها را تعیین می‌کند.
- **بهره‌وری منابع:** درجه‌ای است که در آن منابع سیستم مورد بهره‌برداری قرار می‌گیرند. یک الگوریتم توازن بار مناسب حداکثر بهره‌برداری از منابع را فراهم می‌کند.
- **زمان مهاجرت:** زمانی که جهت مهاجرت یک کار از یک ماشین به هر ماشین دیگر در سیستم طی می‌شود را گویند .جهت بهبود کارایی سیستم ابر این زمان باید حداقل شود.
- **درجه عدم توازن:** این معیار عدمتوازن در میان ماشین-های مجازی را اندازه‌گیری می‌کند.
- **کارایی:** این معیار کارایی سیستم بعد از اجرای الگوریتم توازن بار را اندازه‌گیری می‌کند.
- **مصرف انرژی:** این معیار مقدار انرژی مصرف شده به وسیله تمام گرهها را محاسبه میکند .توازن بار به پرهیز از گرمایش شدید کمک میکند و بنابراین انرژی مصرفی را با توازن بار در میان تمام گره‌ها کاهش می‌دهد.
- **انتشار کربن:** این معیار مقدار کربن تولید شده به وسیله تمام منابع را محاسبه می‌کند. توازن بارنقش مهمی در حداقل کردن این معیار با انتقال بارها از گره‌های فروریز از بار و خاموش کردن آن‌ها را برعهده دارد.

۴- الگوریتم توازن‌بار در رایانش ابری

از نظر اینکه چه موجودیتی فرآیند توازن بار را آغاز می‌کند، الگوریتم-های توازن بار می‌توانند در سه دسته قرار گیرند [۹].

۱) **فرستنده آغازکننده:** اگر اجرای الگوریتم توازن بار به وسیله فرستنده گره سرریز از بار شده آغاز شود.

۲) **گیرنده آغازکننده:** اگر اجرای الگوریتم توازن بار به وسیله گیرنده گره فروریز از بار آغاز شود.

۳) **متقارن** :ترکیب هر دو روش قبلی می‌باشد.

براساس توجه و یا عدم توجه به اطلاعات وضعیت جاری سیستم، الگوریتم‌های توازن بار به سه دسته تقسیم می‌شوند:

۱) الگوریتم توازن بار ایستا[11]

۲) الگوریتم توازن بار پویا [12]

که در ادامه این بخش به بررسی هر کدام از آن‌ها خواهیم پرداخت.

۴-۱- الگوریتم توازن بار ایستا

الگوریتم‌های ایستا به وضعیت جاری سیستم وابسته نیستند و به دانش اولیه از سیستم مانند :نیازمندی‌های منبع کار، زمان ارتباطات، توان یا قدرت پردازشی گره‌های سیستم، حافظه، ظرفیت دستگاههای ذخیره-سازی و غیره نیازمند می‌باشند. در الگوریتم‌های ایستا تصمیمات توازن بار در زمان کامپایل و بر اساس کارایی گره‌های محاسباتی گرفته می‌شود و در خلال زمان اجرا ثابت باقی می‌ماند. در این روش، هنگامی که کار تخصیص می‌یابد هیچ تغییر و یا تخصیص مجددی در زمان اجرا امکانپذیر نمی‌باشد. تعداد کارها در هر گره در الگوریتم‌های توازن بار ایستا ثابت می‌باشد. الگوریتم‌های توازن بار ایستا هیچ اطلاعاتی را در ارتباط با گره‌های محاسباتی جمع‌آوری نمی‌کنند. اختصاص کارها به گره‌های محاسباتی بر اساس پارامترهای زیادی مانند توسعه منبع مورد نیاز، میانگین زمان اجرا، ارتباطات میان فرآیندها و غیره صورت می‌گیرد و این فاکتورها باید قبل از تخصیص اندازه گرفته شوند [۱۰]. مشکل اصلی الگوریتم‌های توازن بار ایستا این است که وضعیت جاری در هنگام تصمیم‌گیری مورد توجه قرار نمی‌گیرد و انتخاب نهایی گره محاسباتی برای تخصیص کار هنگامی که کار ایجاد می‌شود انجام می‌شود و نمی-تواند در خلال فرآیند اجرا تغییر کند.

الگوریتم‌های توازن بار ایستا به دو دسته تقسیم‌بندی می‌شوند:

توازن بار ایستای بهینه: اگر تمام اطلاعات و منابع مربوط به یک سیستم شناخته شده باشند، توازن بار ایستای بهینه می‌تواند انجام شود. در الگوریتم‌های بهینه کنترل کننده مرکزداده اطلاعات مربوط به کارها و منابع را تعیین می‌کند و متوازن کننده بار می‌تواند یک تخصیص بهینه را در زمان معقول انجام دهد [۱۰].

توازن بار ایستای نیمه بهینه: اگر متوازن کننده بار به هر دلیلی نتواند یک تخصیص بهینه را انجام دهد، یک تخصیص نیمه‌بهینه محاسبه خواهد شد. در یک مکانیزم تقریبی، الگوریتم توازن بار بعد از پیدا کردن یک راه حل خوب پایان می‌یابد، به این معنی که آن تمام فضای پاسخ را جستجو نمی‌کند. بعد از آن راه‌حل به وسیله یک تابع عینی ارزیابی می‌شود [۱۰].

۴-۲- الگوریتم‌های توازن بار پویا

الگوریتم‌های توازن بار پویا جهت تصمیم‌گیری توازن بار از اطلاعات وضعیت سیستم در زمان اجرا استفاده می‌کنند و در هنگام مواجه شدن با بارهای پردازشی غیر قابل پیش‌بینی بکار می‌روند. در الگوریتم‌های توازن بار پویا نیاز با دانش اولیه در ارتباط با سیستم نمی‌باشد و تخصیص کارها در زمان اجرا انجام می‌پذیرد. همچنین در الگوریتم‌های توازن بار پویا، کارها در زمان اجرا، بسته به شرایطی که بار از گره‌های محاسباتی با بار سنگین به گره‌های با بار سبک منتقل می‌شود، تخصیص مجدد می‌شوند .لذا مزیت اصلی الگوریتم‌های پویا این است که می‌توانند به طور پیوسته، بر اساس وضعیت جاری سیستم تغییر

کنند. در این شرایط سربار ارتباطات به وجود می‌آید و هنگامی که بارکاری سیستم زیاد می‌شود این سربار بیشتر می‌شود تقریبا تمام الگوریتم‌های پویا از چهار گام پیروی می‌کنند [۱۱]:

۱) **مانیتورکردن بار:** در این گام، بار و وضعیت منابع مانیتور میشود.
۲) **همگام‌سازی:** در این گام اطلاعات وضعیت بار، تبادل میشود.
۳) **معیارهای توازن مجدد:** این گام جهت محاسبه یک توزیع جدید و گرفتن تصمیمات توازن بار براساس این محاسبات ضروری است.
۴) **مهاجرت کار:** در این مرحله انتقال داده‌ها اتفاق می‌افتد .هنگامی که سیستم جهت انتقال کارها تصمیم می‌گیرد این گام اجرا می شود.

الگوریتم‌های توازن بار پویا به دو دسته تقسیم‌بندی می‌شوند که در ادامه این بخش به بررسی آن‌ها خواهیم پرداخت.

۴-۲-۱- الگوریتم توازن بار پویا توزیع شده[۱۳]

در رویکرد توزیع شده، تمام گره‌ها در سیستم، الگوریتم توازن بار را اجرا می‌کنند و وظیفه توازن بار در میان آن‌ها به اشتراک گذاشته می‌شود. عامل بین گره‌های سیستم به دو شکل اشتراکی و غیر اشتراکی انجام می‌گیرد. در حالت اشتراکی، گره‌ها با یکدیگر جهت دستیابی به یک هدف مشترک کار می‌کنند.برای مثال جهت کاهش زمان پاسخ تمام کارها در حالت غیراشتراکی هر گره به صورت مستقل جهت دستیابی به یک هدف محلی کار می‌کند. الگوریتم‌های توازن بار پویا جهت کاهش زمان پاسخ یک کار محلی با طبیعت توزیع شده، معمولا پیام‌های بیشتری از الگوریتم‌های توازن بار توزیع نشده تولید می‌کنند زیرا هر یک از گره‌ها در سیستم نیاز به برقراری ارتباط با همه گره‌های دیگر دارد. مزیت این روش این است که حتی اگر یک یا چند گره در سیستم خراب شود، کل فرآیند توازن بار متوقف نخواهد شد بلکه تنها اندکی بر کارایی سیستم تاثیر خواهد گذاشت. این روش زمانی که گره‌ها به صورت منفرد عمل کرده و تعاملات اندکی با یکدیگر دارند مزیت بیشتری دارد [۱۲].

۴-۲-۲- الگوریتم توازن بار پویا توزیع نشده[۱۴]

الگوریتم‌های توزیع نشده به دو دسته متمرکز و شبه توزیع شده تقسیم-بندی می‌شوند. در حالت متمرکز یک گره مجرد که گره مرکزی نامیده می‌شود الگوریتم‌های توازن بار را اجرا می‌کند و به طور کامل مسئولیت توازن بار را بر عهده دارد. سایر گره‌ها با گره مرکزی تعامل دارند. اما در رویکرد شبه‌توزیع شده، گره‌ها در سیستم، به خوشه‌هایی تقسیم-بندی می‌شوند و هر خوشه به صورت متمرکز عمل می‌کند. گره‌های مرکزی خوشه‌ها، توازن بار در سیستم را بدست می‌دهند. الگوریتم‌های توازن بار پویای متمرکز جهت رسیدن به تصمیم‌گیری، پیام‌های کمتر را دریافت می‌کنند. با این وجود الگوریتم‌های متمرکز می‌توانند در گرهی مرکزی باعث ایجاد گلوگاه در سیستم شوند و همچنین در صورت از کار افتادن گرهی مرکزی کل فرآیند توازن بار متوقف خواهد شد. بنابراین این الگوریتم‌ها بیشتر برای شبکه‌هایی با اندازه کوچک مناسب می‌باشند [۱۲].

۵- پیشینه تحقیق

در این بخش به مروری از الگوریتم‌های توازن بار در محیط رایانش ابری که در چندین سال اخیر توسط پژوهشگران مورد بررسی قرار گرفته است می‌پردازیم.

۵-۱- الگوریتم توازن بار خفه شده[۱۵]

در این الگوریتم متوازن کننده بار لیستی از تمام ماشین‌های مجازی در سیستم را به همراه اطلاعات وضعیت آن‌ها نگهداری می‌کند. این لیست براساس نوع درخواست‌هایی که ماشین‌های مجازی می‌توانند انجام دهند در یک جدول شاخص‌گذاری می‌شوند. به این صورت هنگامی که متوازن‌کننده بار یک درخواست را دریافت می‌کند جدول شاخص‌گذاری شده را جستجو می‌کند. اگر ماشین مجازی مناسب در دسترس باشد آن‌گاه کار به آن ماشین مجازی اختصاص می‌یابد و جدول شاخص‌گذاری شده بروز می‌شود، در غیر این صورت اگر ماشین مجازی مناسب یافت نشود کار در صف انتظار قرار خواهد گرفت [۱۳]. گوش و همکاران در [۱۳] یک رویکرد اولویت‌دار را به منظور بهبود زمان اجرای وظایف براساس الگوریتم توازن بار خفه شده بهبود یافته (PMTA) ارائه کردند. این الگوریتم به منظور تخصیص اجرای وظایف با اولویت بالا و توزیع حجم کار برابر بین چندین ماشین مجازی از یک صف سوئیچینگ استفاده می‌کند. این رویکرد زمان پاسخ و زمان انتظار را در مقایسه با الگوریتم نوبت گردشی بهبود می بخشد، اما ممکن است باعث گرسنگی و زمان پاسخ بالا برای وظایف با اولویت پایین شود.

۵-۲- الگوریتم توزیع متوازن بار جاری اجرایی[۱۶]

همانطور که از نام آن مشخص می‌باشد، این الگوریتم جهت توزیع کردن متوازن بار اجرایی، بر روی ماشین‌های مجازی مختلف تلاش می‌کند. متوازن کننده بار یک جدول شاخص‌گذاری شده از ماشین‌های مجازی را همراه با تعداد درخواست‌هایی که در حال حاضر به هر ماشین مجازی

اختصاص داده شده است را نگهداری می‌کند. اگر یک درخواست جدید از جانب مرکزداده برای اجرا شدن وجود داشته باشد، متوازن کننده بار جدول شاخص‌گذاری شده را برای یافتن یک ماشین مجازی با حداقل میزان بار جستجو می‌کند. اگر بیش از یک ماشین مجازی یافته شود، اولین ماشین مجازی انتخاب شده، و درخواست برای اجرا شدن به آن اختصاص می‌یابد. متوازن‌کننده بار جدول را با افزایش تعداد تخصیص-های ماشین مجازی مذکور بروز می‌کند و زمانی که ماشین مجازی اجرای آن درخواست را به پایان رساند متوازن کننده بار دوباره جدول شاخص‌گذاری شده را با کاهش تعداد تخصیص‌های آن ماشین مجازی بروز می‌کند [۱۵]. کاک‌کار و همکاران [۱۶]، یک رویکرد ترکیبی را برای کاهش زمان پاسخ توسط الگوریتم‌های توزیع متوازن بار جاری اجرایی - توازن‌بار خفه شده در رایانش ابری پیشنهاد کرده‌اند. این رویکرد ترکیبی برخلاف الگوریتم توازن‌بار خفه شده استاندارد که اگر ماشین‌های مجازی مشغول باشند یک را برمی‌گردانند، این رویکرد از یک لیست هش‌مپ برای حفظ درخواست‌های کاربر استفاده می‌کند که تا از طریق این لیست ماشین‌های مجازی در دسترس را جستجو کند و وظایف را به آن اختصاص دهد.

رویکرد مشابه دیگری توسط نویسندگان در [۱۷] پیشنهاد شده است. این رویکرد ترکیبی برای بهینه‌سازی بیشتر منابع است. هدف این رویکرد ترکیبی استفاده از تابع ایجاد میزبان برای کاهش زمان انتظار در صف است. در حالی که هر دو الگوریتم برای کاهش زمان پاسخ مناسب هستند اما در صورت بروز هر گونه خطا هر دو الگوریتم مهاجرت ماشین مجازی را در نظر نمی‌گیرند. در [۱۸] یک رویکرد ترکیبی دیگر که با عنوان تعادل بار ماشین مجازی ترکیبی ارائه شده است. این رویکرد حجم کار و وضعیت فعلی ماشین‌های مجازی را در نظر می‌گیرد تا تصمیم بگیرد که آیا وظایف را به آن‌ها اختصاص دهد یا خیر. همچنین این رویکرد ترکیبی با نگه داشتن فهرستی از کلودلت‌های اختصاص داده شده، می‌تواند ماشین‌های مجازی بارگذاری شده و کم بار را تعیین کند. نتایج آزمایش‌ها نشان می‌دهد که این رویکرد می‌تواند زمان پاسخ را کاهش دهد، اما زمان پردازش مراکز داده این رویکرد در مقایسه با روش‌های پیشین دارای بهبود قابل قبولی نمی‌باشد. بابو و همکاران [۱۹]، یک تکنیک ترکیبی را با ترکیب الگوریتم‌های (توزیع متوازن بار جاری اجرایی - توازن‌بار خفه شده) برای حل مشکل استفاده کم از منابع در رایانش ابری با دستیابی به مقدار مصرف بهینه پیشنهاد کرده‌اند. این تکنیک به سه حالت اشغال، در دسترس بودن و غیره تقسیم می‌شود تا به هر وضعیت ماشین مجازی رسیدگی کند. این رویکرد می‌تواند زمان انتظار، زمان چرخش و هزینه پردازش را کاهش دهد، اما در صورت خرابی ماشین‌های مجازی، تحمل خطای کمی دارد.

۵-۳- الگوریتم Min-Min

این الگوریتم همانند الگوریتم کمینه عمل می‌کند به این صورت که بعد از یافتن کمترین زمان لازم برای اجرای هر کار گره مرکزی کارهایی را که دارای بیشترین زمان لازم برای اجرا هستند را در ابتدا انتخاب می‌کند. این الگوریتم نیز می‌تواند منجر به به وجود آمدن گرسنگی برای کارهایی که زمان اجرای اندکی دارند شود [۲۱]. در [۲۲] یک الگوریتم توازن بار دومرحله‌ای ارائه‌شده است. الگوریتم پیشنهادی، ترکیبی از الگوریتم‌های زمان‌بندی توازن بار فرصت‌طلب و توازن بار Min-Min که می‌تواند کارایی اجرای بهتر و حفظ توازن بار در سیستم را ارائه دهند، می باشد. در این رویکرد، یک عامل به‌طور عمده اطلاعات مربوط به هر گره شرکت‌کننده ظرفیت باقی‌مانده پردازنده، حافظه باقی‌مانده و نرخ انتقال را جمع‌آوری می‌کند. پس از جمع‌آوری همه داده‌ها و ارائه آن‌ها به توازن سیستم کمک می‌شود. نقطه ضعف اصلی این راهکار در سربار اضافی است که در هر بار جمع اوری اطلاعات مربوط به ظرفیت پردازنده و حافظه می باشد که در نمتیجه باعث می‌شود زمان اجرای این راهکار بالا رود.

ارشد علی و همکاران [۲۳] یک رویکرد Min-Min آگاه از منابع را پیشنهاد داده‌اند. این رویکرد که برای تقویت الگوریتم استاندارد Min-Min طراحی شده است. رویکرد پیشنهادی ماتریسی را برای ذخیره وظایف فراهم می‌کند که با در نظر گرفتن زمان تکمیل اجرا وظایف را به منبع مورد نیاز اعلام می‌دارد. در این رویکرد زمان بندی وظایف بسته به دو پارامترحداقل اجرای مورد انتظار و زمان تکمیل وظایف در ماشین مجازی انجام می شود. از محدویت‌های این رویکرد می‌توان به در نظر نگرفتن بار فعلی و به‌روز شده ماشین‌های مجازی در زمان فرآیند تخصیص وظایف اشاره کرد. پاتل و همکاران [۲۴] الگوریتم دیگری با عنوان بار منبع Min-Min پیشنهاد کردند. این الگوریتم وظایف را با حداقل زمان اجرا پیدا می‌کند سپس به ماشین مجازی که کمترین زمان تکمیل را دارند اختصاص می‌دهد. نتایج شبیه‌سازی نشان‌دهنده این است که الگوریتم پیشنهادی هزینه استفاده و توان عملیاتی سیستم را کاهش می‌دهد، اما اولویت های کار را در نظر نمی گیرد.

آروک‌کیم و همکاران [۲۵] یک الگوریتم Min-Min متعادل کننده بار مبتنی بر منبع را پیشنهاد کردند که برای کاهش زمان تکمیل وظایف[۱۷] و متعادل کردن حجم کار در ماشین‌های مجازی طراحی شده است. روش پیشنهادی مقدار زمان تکمیل وظایف را پس از تخصیص به منبع محاسبه می‌کند و سپس این مقدار را به عنوان مقدار آستانه در نظر می‌گیرد. نتایج نشان می‌دهد که حداکثر زمان تکمیل وظایف برای الگوریتم پیشنهادی برابر ۷ میلی‌ثانیه می‌باشد اما این روش اولویت وظایف و پارامترهای مهمی مانند کیفیت سطح خدمات و مهلت را در نظر نمی‌گیرد.

۵-۴- الگوریتم نوبت گردشی[۱۸]

این الگوریتم یک الگوریتم توازن بار ایستا است که از روش نوبت گردشی برای اختصاص کارها استفاده می‌کند. این الگوریتم گره اول را به صورت تصادفی انتخاب می‌کند و سپس به شیوه نوبت گردشی کارها را به تمام گره‌های دیگر اختصاص می‌دهد. کارها به ترتیب و بدون هیچ گونه اولویتی به پردازنده‌ها تخصیص می‌یابند، به هر پردازنده به صورت یکسان برش زمانی و ترتیب چرخشی نسبت داده می‌شود .اگر پردازشی قبل از انقضای زمان پردازنده مختص به خود کامل نشود، پردازش لغو می‌شود و به پردازش بعدی منتظر در صف منقل می‌شود. این الگوریتم به دلیل وجود بار کاری غیریکنواخت و مشخص نبودن پیش‌بینی زمان اجرای کارها در آن الگوریتم مناسبی برای رایانش ابری نمی‌باشد. زیرا زمان اجرای هیچکدام از کارها از ابتدا مشخص نمی‌باشد، برخی گره‌ها دارای بار سنگین و برخی دیگر بار اندکی را خواهند داشت. در این الگوریتم وزن‌های دقیقی به هر گره اختصاص داده می‌شود. براساس وزن اختصاص یافته هر گره تعداد مناسبی از کارها را دریافت می‌کند، اگر تخصیص وزن‌ها به گره‌ها به صورت یکسان باشد تمام گره‌ها ترافیک یکسانی را دریافت می‌کنند [۲۶]. تی‌لونگ و همکارانش [۲۷] یک الگوریتم بهینه‌سازی شده را برای کاهش زمان پاسخ در محیط کلود آنالیست پیشنهاد کردند. الگوریتم پیشنهادی زمان پاسخ و زمان انتظار را برای هر فرآیند محاسبه می‌نماید و سپس در مورد فرآیند زمان‌بندی آن‌ها تصمیم‌گیری می‌کند. از مزایای این رویکرد پیشنهادی کاهش زمان پاسخ‌دهی می‌باشد. از معایب آن می‌توان به کوانتوم زمانی اشاره کرد به این دلیل که چون الگوریتم پیشنهادی میتنی بر الگوریتم نوبت گردشی می‌باشد این امر باعث می‌شود که الگوریتم مورد نظر برای محیط‌های ابری پویا بهینه و مناسب نباشد.

کاورآو و همکارانش [۲۸]، رویکرد الگوریتم ژنتیک را برای بهینه کردن توازن بار در محیط ابر پیشنهاد داده‌اند. هدف این رویکرد پیشنهادی ارائه یک تعادل بار کارآمد است که می‌تواند قابلیت مراکز داده را بهبود بخشد. در رویکرد پیشنهادی نویسندگان برای بهینه‌سازی چالش الگوریتم نوبت گردشی از تابع درهمساز (هش) استفاده نموده‌اند، این تابع تمام درخواست‌های محیط ابر را مورد جستجوی قرار می‌دهد و آن‌ها را به تمام ماشین های مجازی که قابلیت در دسترس بودن را دارند اختصاص می‌دهد و سپس توسط الگوریتم ژنتیک بهینه‌ترین ماشین مجازی که می‌تواند وظایف بیشتری را اجرا کند انتخاب می‌شود. نتایج نشبیه‌سازی نشان می‌دهد که رویکرد پیشنهادی زمان پاسخگویی سرورها را به میزان قابل توجهی کاهش می دهد. اما از طرفی چون الگوریتم ژنتیک نیاز به زمان بیشتری برای فضا جستجو دارد این امر باعث پیچیدگی روش پیشنهادی می‌گردد.

ریچاریا و همکارانش [۲۹]، یک رویکرد پیشنهادی ترکیبی از خط مشی اولویت - الگوریتم نوبت گردشی را پیشنهاد کرده‌اند. در این رویکرد ترکیبی، الگوریتم نوبت گردشی بهبود یافته شامل دو پردازنده است: از یک پردازنده کوچک برای محاسبه برش زمانی فرآیندها استفاده می‌شود و یک پردازنده اصلی که در آن فرآیندها بر اساس زمان‌های پشت سر هم که به عنوان مقدار اولویت در نظر گرفته می‌شوند و به صورت صعودی مرتب می شوند. هدف این رویکرد کاهش زمان پاسخ است، اما هیچ تست عملکردی برای استنباط نتایج وجود ندارد.

۵-۵- الگوریتم زنبور عسل[۱۹]

این مفهوم برگرفته شده از رفتار زنبورعسل برای جستجو و جمع‌آوری ماده خوراکی می‌باشد. دسته‌ای از زنبورها به نام زنبورهای جستجوگر برای غذا وجود دارند که برای منابع غذایی جستجو می‌کنند و پس از یافتن منبع غذایی به کندو بازگشته و شروع به تبلیغ کردن این منبع غذایی با استفاده از یک نوع رقص مخصوص به نام رقص جنبش می‌کنند. نمایش این رقص اطلاعات مربوط به کیفیت، کمیت و یا حتی فاصله‌ی آن منبع غذایی تا کندو را به سایر زنبورها می‌دهد. این یک روش توازن بار غیرمتمرکز ایده گرفته شده از طبیعت می‌باشد که به دستیابی توازن بار در میان ماشین‌های مجازی مختلف در محیط رایانش ابری کمک می‌کند.میزان بارکاری هر ماشین مجازی مورد توجه قرار گرفته و سپس تعیین می‌شود که آیا ماشین مجازی در وضعیت سرریز از بار یا در وضعیت فروریز از بار و یا در وضعیت متعادل قرار دارد. سپس ماشین‌های مجازی به نسبت میزان بار موجودشان گروه‌بندی می‌شوند وکارها از ماشین‌های مجازی سرریز از بار شده حذف و به یک ماشین مجازی با بار سبک زمانبندی می‌شوند.کارهای قبلی حذف شده در پیدا کردن ماشین مجازی با بار سبک مفید خواهند بود .این کارها همانند زنبورهای عسل پیرو خواهند بود. تکنیک توازن بار الهام گرفته شده از رفتار زنبور عسل زمان واکنش ماشین مجازی و همچنین زمان انتظار کار را کاهش می‌دهد [۳۰].

احسانی مقدم و همکاران [۳۱] الگوریتم تعادل بار را مبتنی بر رفتار زنبور عسل برای کاهش زمان جستجو و تخصیص وظایف را در ابر عمومی مورد بررسی قرار داده‌اند. در این الگوریتم پس از محاسبه بار هر ماشین مجازی، ابر عمومی به سه بخش به نام های ابر کم بار شده ، ابر متعادل و ابر بیش از حد بارگذاری شده تقسیم می‌شود، سپس تفاوت مقدار بین بارها توسط الگوریتم پیشنهادی محاسبه می‌شود. بر این اساس اگر این مقدار بیشتر از صفر باشد، هیچ حرکت ماشین مجازی از یک بخش به بخش دیگر مورد نیاز نیست. این روش سرعت و هزینه را به عنوان یک اولویت اصلی برای ماشین های مجازی در نظر می گیرد، اما راه حلی برای دو اولویت مشابه ارائه نمی‌دهد.

۵-۶- الگوریتم ژنتیک[۲۰]

الگوریتم ژنتیک یکی از الگوریتم‌های جستجوی تصادفی است که ایده آن برگرفته از طبیعت می‌باشد. این الگوریتم در حل مسائل بهینه‌سازی کاربرد دارد. در طبیعت از ترکیب کروموزوم‌های بهتر، نسل‌های بهتری پدید می‌آیند. دراین بین گاهی اوقات جهش‌هایی نیز در کروموزوم‌ها روی می‌دهد که ممکن است باعث بهتر شدن نسل بعدی شوند. الگوریتم ژنتیک نیز با استفاده از این ایده اقدام به حل مسائل می‌کند.یک الگوریتم ژنتیک برای حل یک مسئله، مجموعه‌ی بسیار بزرگی از راه-حل‌های ممکن را تولید می‌کند. هر یک از این راه‌حل‌ها با استفاده از یک تابع تناسب مورد ارزیابی قرار می‌گیرند. آنگاه تعدادی از بهترین راه‌حل‌ها باعث تولید راه حل‌های جدید می‌شوند. این باعث تکامل راه-حل‌ها می‌گردد [۳۲]. بدین‌ترتیب فضای جستجو در جهتی تکامل پیدا می‌کند که به راه‌حل مطلوب برسد. در صورت انتخاب صحیح پارامترها این روش می‌تواند بسیار مؤثر عمل کند. برای ساخت یک الگوریتم ژنتیک، تعدادی مراحل وجود دارد. عبارتند از:

۱) **شروع** : جمعیتی تصادفی متشکل از N کروموزوم را به-صورت تصادفی تولید می‌کند و راه‌حل‌های مناسب برای مسئله را تولید می‌نماید.
۲) **برازندگی:** ارزیابی برازندگی برای هر کروموزوم x در جمعیت ژنتیکی توسط تابع f(x) انجام می‌گیرد.
۳) **جمعیت جدید:** یک جمعیت جدید را ایجاد می‌کند و گام‌ها را تا کامل شدن جمعیت جدید نشان می‌دهد.
۴) **انتخاب**: دو کروموزوم والدین را از یک جمعیت بر پایه برازندگیشان انتخاب می‌کند. بهترین برازندگی، شانس بیشتری برای انتخاب دارد.
۵) **برش:** با یک احتمال برش، اولاد جدید از روی والدین تشکیل می‌شوند.
۶) **جهش:** با یک احتمال جهش، اولاد جدید از هر مکانی قابل تغییر هستند.
۷) **پذیرفتن:** قرار دادن اولاد جدید در یک جمعیت جدید.
۸) **جایگذاری:** بکار بردن جمعیت تولید شده جدید برای اجرای مجدد الگوریتم.
۹) **تست:** اگر شرط نهایی برآورده شود، متوقف می‌شود و بهترین راه حل، برگردانده می‌شود.
۱۰) **حلقه:** رفتن به گام دوم.

وانگ و همکاران [۳۳]، به زمانبندی وظایف و توازن بار مبتنی بر الگوریتم ژنتیک در محاسبات ابری پرداختند. زمانبندی وظایف یکی از مهمترین مسائل در پلتفرم ابر است. تعداد کاربران و حجم دادهها بسیار بزرگ است .درخواست‌های به اشتراک‌گذاری دارایی و استفاده مجدد از آنها ضروری است. مکانیزم مؤثر زمانبندی وظایف نیازمندی‌های کاربران را برآورده و باعث بهبود استفاده از منابع می‌شود. برای حل این مسئله، با درنظر گرفتن خواص جدید محاسبات ابری و الگوریتم ژنتیک پایه یک الگوریتم زمانبندی جدید مبتنی بر الگوریتم ژنتیک بر پایه زمانبندی و توازن بار ارائه شده است. این استراتژی نه تنها باعث زمانبندی وظایف و کاهش زمان تکمیل وظایف می‌شود بلکه توازن بار بین گره‌ها را رضایت‌بخش می‌کند. همچنین در این روش از الگوریتم حریصانه برای مقداردهی اولیه جمعیت، پراکندگی توصیف بارهای شدید بین گره‌ها و از طریق شبیه‌سازی AGA با JLGA تابع برازندگی وزن‌دار استفاده شده است. از لحاظ عملکرد روش مقایسه شده است که اعتبارسنجی و اثربخشی آن را اثبات می‌کند.

پیلواره و همکاران [۳۴]، یک روش جدید برای بهبود عملکرد توازن بار با استفاده از الگوریتم ژنتیک در محاسبات ابری ارائه دادند. همانطور که محاسبات ابری با اتصال از طریق شبکه با سرورها انجام می شود بنابراین مسائل زیادی وجود دارد که باید حل شوند .تعادل بار، مسئله اصلی در مورد ابر است که باید حل شود. تکنیک‌های مختلف برای بهبود تعادل بار در محاسبات ابر استفاده می‌شود. در میان آنها تکنیک های مختلف مبتنی بر الگوریتم ژنتیک بهتر عمل می کنند. الگوریتم ژنتیک از انتخاب تصادفی پردازنده‌ها به عنوان ورودی استفاده می‌کند و سپس پردازش را انجام می‌دهد. قبلاً پردازنده‌ها و وظایف اولویت‌بندی می‌شدند که واقعی نبود. بنابراین برای بهبود کارایی الگوریتم ژنتیک پردازنده‌های ورودی ابتدا به الگوریتم اولویتی داده می‌شوند که ماتریس کمترین مربعات لگاریتمی در این روش پیشنهاد شده است. مسئله بیکاری و قحطی‌زدگی با الگوریتم پیشنهادی حل شده است.

در تحقیق ارائه شده در [۳۵]، کارها به ترتیب سطح جریان کاری‌شان درون کروموزم‌ها مرتب می‌کنند و عملگرهای تقطیع و جهش را بر اساس نمایش سطحی انتخاب می کنند. در این راهکار عملگرهای انتخاب، جهش و تقطیع متفاوتی باهم مقایسه شده‌اند. در این پژوهش به این نتیجه رسیده‌اند که عملگر تقطیع چرخه‌ای برای کاهش زمان خاتمه از همه آن‌ها بهتر عمل می‌کند. به طور مشابه عملگر ایجاد توازن مجدد به‌عنوان بهترین عملگر جهش در بین عملگرهای حرکت، تعویض، حرکت و تعویض، و ایجاد توازن مجدد مشخص شد. بعد از مقایسه پنج عملگر انتخاب تصادفی، بهترین، رتبه‌بندی خطی، تورنومنت دودویی و N تورنومنت، انتخاب تورنومنت آن‌ها به این نتیجه رسیدند که انتخاب N تورنومنت از همه بهتر است.

۵-۷- الگوریتم بهینه‌سازی ذرات ازدحام[۲۱]

الگوریتم بهینه‌سازی ازدحام ذرات یک الگوریتم اکتشافی است که بر روی رفتار اجتماعی ذرات در یک جمعیت تمرکز دارد. هر ذره بهترین موقعیت محلی و بهترین موقعیت عمومی را در کل جمعیت بدست می‌آورد. در این روش هر یک از کارها به عنوان یک تک ذره در یک اجتماع بزرگ در نظر گرفته می‌شود. انتخاب ذرات به وسیله سرعت، جهت و بزرگی آنها کنترل می‌شود. در این روش جمعیت‌های مختلفی وجود دارد. هر جمعیت شامل ذراتی با بزرگی و جهت‌های گوناگون می‌باشد. ذرات همچنین دارای یک مقدار برازندگی می‌باشند که در هر تکرار ارزیابی و بهینه می‌شود. کارایی ذرات به وسیله مقدار برازندگی آن‌ها ارزیابی می‌شود [۳۶]. در [۳۷] یک راهکار مبتنی بر بهینه سازی ازدحام به منظور توازن بار و تخصیص منابع ارائه گردیده است، در این راهکار وظایف در قالب ذرات شبیه‌سازی شده‌اند، به این صورت که ابتدا با بهره‌گیری از یک زمانبند کلیه وظایف در یک فضای جستجو به منظور یافتن بهترین سروری که بتواند وظایف را اجرا کند، قرار می‌گیرند. به هر یک از ذرات به‌صورت تصادفی یک سرعت و مکان تصادفی داده می‌شود که در هر بار اجرا، به روزرسانی می‌شود، در انتها این نتیجه حاصل شده که راهکار پیشنهادی می‌توان بخوبی میزان انرژی را کاهش دهد. اما یکی از مشکلات اصلی این راهکار عدم بهینه‌سازی سرورها در طی فرایند اجرا می باشند، به این صورت که امکان دارد در طی فرایند اجرا، تعدادی از سرورها در اثر مشکلات سخت افزاری یا نرم‌افزاری دچار شکست شوند، در حالیکه در این تحقیق هیچگونه تدبیری در این حالت ارائه نشده است ئو درنتیجه می‌تواند باعث ازدحام و افزایش زمان اجرا در سرورها شود.

لو و همکاران [۳۸]، به توازن بار در محیط محاسبات ابری با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات پرداختند. مهاجرت ماشین مجازی یک تکنیک برای رسیدن سیستم به توازن بار در محیط محاسبات ابری است تا به وسیله آن یک ماشین مجازی فعال از یک میزبان فیزیکی به میزبان دیگر منتقل شود. روش پیشنهادی برای کاهش زمان بیکاری ماشین‌های مجازی استفاده می‌شود. در این مقاله وظیفه مبتنی بر سیستم توازن بار با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات پیشنهاد شده است تا سیستم از طریق انتقال وظایف اضافی از ماشین مجازی با سربار بالا، توازن بار را ایجاد کند. برای شبیه‌سازی از نرم‌افزار متلب استفاده شده است. نتایج شبیه‌سازی نشان می‌دهد روش پیشنهادی باعث کاهش زمان تکمیل وظایف شده و نتایج با روشهای کلاسیک مقایسه شده است. علاوه بر آن، کار ماشین‌های مجازی در حین فرایند مهاجرت متوقف نمی‌شود. روش پیشنهادی باعث حذف ماشین مجازی بیکار شده است و کیفیت سرویس مشتریان ابر افزایش یافته است.

در [۳۹]، ترکیبی از الگوریتم بهینه‌سازی ازدحام ذرات و الگوریتم جستجوی محلی شبیه‌سازی گرانشی[۲۲](GELS) برای زمان‌بندی کارهای مستقل در محاسبات گرید پیشنهاد شده است. GELS یک الگوریتم جستجوی محلی است که برای بهبود نتایج به دست آمده بعد از الگوریتم بهینه‌سازی ازدحام ذرات و جلوگیری از افتادن در جواب بهینه محلی استفاده می‌شود. الگوریتم GELS نتایج به دست آمده از الگوریتم بهینه‌سازی ازدحام ذرات را برای یافتن بهترین راه‌حل بررسی می‌کند و فضای جستجو را به‌صورت تصادفی کاوش نمی‌کند. نتایج آزمایش‌ها نشان داد که الگوریتم PSO-GELS در مقایسه با الگوریتم شبیه‌سازی حرارت، برای ۴۰۰ کار و صفر منبع زمان خاتمه کمتری در حدود ۲۹,۲ درصد داشته است.

۵-۸- الگوریتم کلونی مورچه[۲۳]

بهینه‌سازی به روش کلونی مورچگان یک روش فرااکتشافی است که در آن یک کلونی از مورچه های مصنوعی، در پیدا کردن جواب‌های بهینه برای مسائل پیچیده، شرکت می‌کنند. مشارکت، یک جزء کلیدی در طراحی الگوریتم‌های مورچگان می‌باشد. در این روش منابع محاسباتی به یک مجموعه از عامل‌های سبتاً ساده (مورچه های مصنوعی) که به‌صورت غیرمستقیم (ارتباط استیگمرجی) با یکدیگر ارتباط برقرار می-کنند، تخصیص داده می‌شود. .یک مورچه مصنوعی در ACO یک رویه سازنده تصادفی است که یک جواب را توسط افزودن اجزای جواب به یک جواب جزئی می‌سازد. بنابراین ازاین راهکار می‌توان بر هر مسأله بهینه‌سازی ترکیبی که برای آن بتوان یک راه حل ابتکاری تعریف کرد، بکار برد. مانند الگوریتم‌های ژنتیک، الگوریتم‌های کلونی مورچگان بر روی مجموعه‌ای از جواب‌ها عمل می‌کنند که به آن جمعیت می‌گویند. جمعیت یک کلونی مورچگان را تداعی می‌کند و هر مورچه عضو این جمعیت معرف یک راه حل یا جواب است.

در الگوریتم‌های کلونی مورچگان، جواب‌های جدید در هر مرحله به‌صورت تصادفی انتخاب می‌شوند، برخلاف الگوریتم‌های ژنتیک که در هر مرحله، جواب‌های جدید از روی جواب‌های موجود ساخته می‌شوند. اولین الگوریتم کلونی مورچگان با نام سیستم مورچه‌ها توسعه یافت که رویکرد جدیدی برای حل مسایل بهینه‌سازی ترکیبیاتی بود. در اصل سیستم مورچه ها از مسأله فروشنده دوره‌گرد به‌عنوان مثال استفاده کرده است. در این الگوریتم فرومون مصنوعی $Z(i,j)$ برای هر کمان (i,j) میان شهر i و شهر j تعریف شده است. مقدار $Z(i,j)$ نشان دهنده میزان تمایل به ملاقات شهر j بلافاصله بعد از شهر i می‌باشد. هرمورچه یک تور که از یک شهر تصادفی شروع می‌شود و به‌صورت تصادفی شهرهای دیگر را، یکی پس از دیگری طی می‌کند را می‌سازد. احتمال اینکه، مورچه‌ای که در شهر i قرار دارد، شهر j را بلافاصله بعد از شهر i ملاقات کند برابر است با [۴۰]:

$$(i,j) = \begin{cases} \frac{Z(i,j)^{\lambda}\eta(i,j)^{\beta}}{\sum_{k\epsilon Z(i,k)}\lambda\eta(i,k)^{\beta}} & if\ j \in V \\ 0 & otherwise \end{cases} \quad (۱)$$

که V مجموعه شهرهایی است که تاکنون توسط مورچه ملاقات نشده-اند، $\eta(i,j)$ مقدار اکتشافی تعریف شده برای مسأله، βوλ نیز پارامترهای تعین کننده میزان اهمیت فرومون‌ها و میزان اکتشافی هستند. بعد از اینکه تمام مورچه های جمعیت، جواب خود را ساختند، دنباله فرومون‌ها به‌صورت زیر به‌روزرسانی می‌شود:

$$Z(i,j) = P \times Z(i,j) + \sum_{k\epsilon p}\frac{W(i,j,k)}{Z(k)} \quad (۲)$$

که p جمعیت، $Z(k)$ مجموع فاصله مسأله فروشنده دوره‌گرد که توسط مورچه k ساخته شده است. در معادله فوق $Z(i,j)$ نشان دهنده میزان تمایل به ملاقات شهر j بلافاصله بعد از شهر i می‌باشد، همچنین نحوه به روزرسانی دنباله فرومون‌ها در [۳۱] به‌طور دقیقتر بیان شده است. در ادامه آمده که اگر $w(i,j,k) = 1$ آنگاه شهر j بعد از شهر iام در جواب k قرار می‌گیرد، در غیر این صورت $w(i,j,k) = 0$ خواهد بود. در واقع قسمت دوم رابطه فوق، نشان دهنده مشاهده فرمون توسط مورچه‌ها است. بنابراین اگر شهر j بلافاصله بعد از شهر iام در جواب‌های با فاصله کوتاهتر قرار گیرد، فرومون z(i,j) قوی و قوی‌تر می‌شود. به عبارت دیگر احتمال ملاقات شهر jام بلافاصله بعد از شهر iام افزایش می‌یابد [۴۰].

هدف اصلی از مقاله [۴۱] توسعه یک الگوریتم توازن بار مؤثر با استفاده از روش بهینه‌سازی کلونی مورچه ها جهت به حداکثر رساندن یا به حداقل رساندن پارامترهای عملکرد مختلف مانند بار پردازنده، ظرفیت حافظه، تأخیر و یا بار شبکه برای ابرها در اندازه مختلف، است. هر چند که این راهکار در به حداکثر رساندن عملکرد پردازنده و ظرفیت حافظه موفق بوده و از طرفی میزان تأخیر نیز کاهش پیدا کرده است اما همزمان با آن میزان مصرف انرژی بشدت بالا رفته است بطوریکه در این زمینه این راهکار جزء پرمصرف‌ترین روشهای توازن بار در ابر می‌باشد. چن و همکارانش در [۴۲]، یک الگوریتم زمان‌بندی برای زمان‌بندی کارها و توازن بار در ابر ارائه دادند. در این الگوریتم تعداد مورچه هابه طور دلخواه انتخاب شده است. هر مورچه بر اساس تابع احتمال و تعدادی از محدودیت ها، یک جواب برای زمان‌بندی کارها ارائه می‌دهد. ترتیب تخصیص کارها به منابع مبتنی بر این محدودیت ها است. الگوریتم پیشنهادی بر اساس احتمال تعریف‌شده، برای هر کار یک منبع انتخاب می‌کند. همچنین هر مورچه به‌صورت ترتیبی و در N گام راه حل را ایجاد می‌کند، که در آن N تعداد کارها می‌باشد. در طول هر تکرار مورچه ها به‌صورت موازی B راه حل را ایجاد می‌کنند که در آن B تعداد مورچه ها است. بعد از تخصیص منبع به هر کار به‌روزرسانی محلی فرمون و در انتهای هر تکرار به‌روزرسانی عمومی فرمون صورت می گیرد. نویسندگان در این الگوریتم قابلیت اطمینان، زمان و هزینه را به‌عنوان پارامترهای کیفیت خدمات در نظر گرفته‌اند. در این الگوریتم کاربران مجاز هستند تا محدودیت‌های کیفیت خدمات را برای تضمین کیفیت زمان‌بندی تعریف کنند. همچنین در این الگوریتم، تابع هدف بهینه‌سازی الگوریتم به تعریف کاربر بستگی دارد. در الگوریتم پیشنهادی هفت اکتشاف و هفت مقدار برای فرمون تعریف‌شده است. اکتشاف‌ها با استفاده از یک الگوی تطبیقی و بر اساس مقدار فرمون‌ها توسط مورچه های مصنوعی انتخاب شده است. این الگوریتم هزینه را به طور قابل توجه کاهش داده است

یک الگوریتم کلونی مورچگان چند هدفه برای مسئله جایابی ماشین‌های مجازی در [۴۳] ارائه‌شده است. هدف اصلی الگوریتم پیشنهاد شده آن ها، به دست آوردن مجموعه ای از راه حل های غیر مغلوب (مجموعه پرتوای) است که به‌طور هم‌زمان اتلاف کلی منابع و مصرف توان را به حداقل رساند. این الگوریتم بر اساس سیستم کلونی مورچگان است. در فاز مقدار دهی اولیه، پارامترها مقدار دهی می‌شوند و تمام مسیرهای فرمون به τ_0 تنظیم می‌شوند. در بخش تکرار شونده، هر مورچه تمام درخواست‌های ماشین مجازی را دریافت می‌کند، یک سرور فیزیکی را معرفی می‌کند و شروع به تخصیص ماشین‌های مجازی به میزبان ها می‌نماید. این تخصیص، با استفاده از یک قاعده شبه تصادفی متناسب به دست می آید. این قاعده بر اساس اطلاعاتی درباره غلظت فرمون فعلی در مسیر حرکت است و یک روش اکتشافی است که مورچه ها را در جهت انتخاب مناسب ترین VMها راهنمائی می‌کند. به روزرسانی فرمون محلی هنگامی انجام می‌شود که یک مورچه مصنوعی یک حرکتی داشته باشد. سپس، تمام مورچه ها راه حل های خود را می سازند، یک به روزرسانی سراسری با یک راه حل از مجموعه پرتو فعلی انجام می‌شود.

این الگوریتم از نظر عملکرد و مقیاس پذیری با الگوریتم ژنتیک چند هدفه ، الگوریتم کلونی مورچگان تک هدفه (معرفی شده در مرجع [۴۲]) و الگوریتم FFD[۲۴] تک هدفه (معرفی شده در مرجع [۴۳]) مقایسه شد. هر یک از الگوریتم‌ها تنظیمات خاص خود را دارند، و باید توجه داشت که تنظیماتی که برای متغیرهای مختلف در نظر می گیریم تأثیر مستقیم روی عملکرد الگوریتم دارند. الگوریتم VMPACS[۲۵] دقیقاً از این نکته بهره برده و تنظیمات خاصی را برای متغیرها با درنظر گرفتن هدف مسئله انتخاب می‌کند که مقدار این تنظیمات را با آزمایش های مقدماتی به دست آورده است. در این روش به جهت سادگی، آنها

محیط سرور همگن را برای روش خود شبیه‌سازی کردند اما روش ارائه‌شده برای شرایط محیط سرور ناهمگن نیز قابل بهره برداری است. به این معنی که تعداد سرورها به منظور حمایت از بدترین سناریوی جایابی ماشین مجازی ، یعنی جائیکه تنها یک ماشین مجازی به هر سرور فیزیکی تخصیص داده می‌شود، تنظیم شد. در این روش آنها همبستگی خطی مصرف حافظه و CPU را برای هر نمونه تعریف کردند و از روش پیشنهادی در منبع [۴۳] برای تولید سری تصادفی از مصرف حافظه و CPU در آزمایش ها برای مواردی که چندین همبستگی خطی در آزمایش وجود دارد استفاده شد. بعد از اینکه الگوریتم جایابی ماشین مجازی کار خود را به پایان رساند چنانچه نتیجه ی الگوریتم چند راه حل غیر مغلوب باشد، الگوریتم به طور تصادفی یک راه حل از مجموعه راه حل های برگزیده را انتخاب می‌کند.

جدول ۱: مقایسه الگوریتم‌های تعادل بار در فضای رایانش ابری

مرجع	سال انتشار	الگوریتم	نقاط قوت	نقاط ضعف	ابزار مورد استفاده
[۱۳]	۲۰۱۶	الگوریتم توازن بار خفه شده بهبود یافته (PMTA)	توزیع برابر حجم کار بر خلاف الگوریتم نوبت گردشی کاهش دادن زمان پاسخگویی بهبود کیفیت خدمات	باعث گرسنگی و زمان پاسخ بالا برای وظایف با اولویت پایین شود.	کلودسیم
[۳۱]	۲۰۱۸	الگوریتم کلونی زنبور عسل	کاهش فرآیند جستجو برای تخصیص کار درخواست دادن مهاجرت ماشین مجازی مناسب برای محیط‌های ناهمگن	در نظر نگرفتن اولویت برای دو ماشین مجازی که دارای اولویت برابر هستند .	کلودسیم
[۱۹]	**۲۰۱۷**	جاری بار متوازن رویکرد ترکیبی توزیع اجرایی - توازن‌بار خفه شده	کاهش میانگین زمان پاسخگویی از بارگذاری بیش از حد موقعیت‌ها ماشین‌های مجازی جلوگیری می‌کند.	مهاجرت ماشین مجازی را در صورت خرابی در نظر نمی گیرد.	کلودسیم
[۳۴]	**۲۰۱۵**	الگوریتم ژنتیک	حداقل زمان صرف شده برای تکمیل درخواست‌های کاربر زمان‌بندی کارآمد در بین ماشین‌های مجازی مختلف	کاهش کارایی با افزایش فضای جستجو در نظر نگرفتن اولویت وظایف	کلود آنالیست
[۴۱]	**۲۰۱۲**	الگوریتم کلونی مورچه	مناسب برای محیط‌های پویا کاهش هزینه تاخیر	عدم تست عملکرد افرایش میزان مصرف انرژی	متلب
[۳۶]	**۲۰۱۲**	الگوریتم بهینه‌سازی ازدحام ذرات	تولید وظایف بهینه پاسخ سریع برای تکمیل فرآیند اجرای وظایف کاهش زمان اجرا کاهش میزان مصرف انرژی	عدم بهینه‌سازی سرورها بروز شکست سرورها در اثر مشکلات سخت افزاری یا نرم‌افزاری افزایش زمان اجرا در سرورها	گریدسیم

۶- نتیجه‌گیری و کارهای آینده

توازن‌بار یک جنبه مهم در زمینه رایانش ابری برای افزایش توزیع بار کاری و استفاده کارآمد از منابع محسوب می‌شود. در این مقاله، الگوریتم‌های توازن بار متعددی از جمله الگوریتم‌های استاتیک، الگوریتم‌های دینامیک و الگوریتم‌های الهام‌گرفته از طبیعت برای بهینه-سازی و متعادل کردن یار در فضای ابری مورد بررسی قرار گرفت. در ادامه این مقاله روش‌های تعادل بار از طریق تحلیل مقایسه‌ای الگوریتم-های پیشنهادی توسط محققان در چند سال اخیر مورد بحث قرار گرفت و مزایا و معایب هر کدام مشخص گردید. تاکنون توسط پژوهشگران رویکردها و الگوریتم‌های زیادی برای حل مسائل مربوط به تعادل بار پیشنهاد شده است، اما هنوز برخی از مسائل مانند مهاجرت ماشین‌های مجازی، مسائل مربوط به تحمل خطا در فضای ابری وجود دارد، که به طور کامل برطرف نشده است. این مقاله می‌تواند بستری مناسب را برای محققان فراهم کند تا الگوریتم‌های متعادل کننده بار هوشمند و کارآمد را برای محیط‌های ابری توسعه دهند. نتایج نشان داد که الگوریتم‌های الهام گرفته از طبیعت مانند الگوریتم کلونی مورچه، الگوریتم ژنتیک و الگوریتم بهینه‌سازی ازدحام ذرات بیشترین کاربرد در

متعادل کردن بار در فضای ابری داشته‌اند و همچنین تا به امروز محیط شبیه‌ساز کلودسیم [۴۴] بیشترین کاربرد را برای بهینه‌سازی الگوریتم-های تعادل بار در فضای ابری داشته است.

مراجع

[1] Rashid A, Chaturvedi A. Cloud computing characteristics and services: a brief review. International Journal of Computer Sciences and Engineering. 2019 Feb;7(2):421-6.

[2] Bokhari MU, Makki Q, Tamandani YK. A survey on cloud computing. InBig Data Analytics: Proceedings of CSI 2015 2018 (pp. 149-164). Springer Singapore.

[3] Kumar M, Sharma SC, Goel A, Singh SP. A comprehensive survey for scheduling techniques in cloud computing. Journal of Network and Computer Applications. 2019 Oct 1;143:1-33.

[4] Arockiam L, Monikandan S, Parthasarathy G. Cloud computing: a survey. Journal of Computer and Communication Technology: Vol. 2017 Jan;8(1):4.

[5] Lin CC, Liu P, Wu JJ. Energy-efficient virtual machine provision algorithms for cloud systems. In2011 Fourth IEEE International Conference on Utility and Cloud Computing 2011 Dec 5 (pp. 81-88). IEEE.

[6] Ghomi EJ, Rahmani AM, Qader NN. Load-balancing algorithms in cloud computing: A survey. Journal of Network and Computer Applications. 2017 Jun 15;88:50-71.

[7] Thakur A, Goraya MS. A taxonomic survey on load balancing in cloud. Journal of Network and Computer Applications. 2017 Nov 15;98:43-57.

[8] Zhang J, Yu FR, Wang S, Huang T, Liu Z, Liu Y. Load balancing in data center networks: A survey. IEEE Communications Surveys & Tutorials. 2018 Mar 15;20(3):2324-52.

[9] Shafiq DA, Jhanjhi NZ, Abdullah A. Load balancing techniques in cloud computing environment: A review. Journal of King Saud University-Computer and Information Sciences. 2022 Jul 1;34(7):3910-33.

[10] Deepa T, Cheelu D. A comparative study of static and dynamic load balancing algorithms in cloud computing. In2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS) 2017 Aug 1 (pp. 3375-3378). IEEE.

[11] Aslam S, Shah MA. Load balancing algorithms in cloud computing: A survey of modern techniques. In2015 National software engineering conference (NSEC) 2015 Dec 17 (pp. 30-35). IEEE.

[12] Shoja H, Nahid H, Azizi R. A comparative survey on load balancing algorithms in cloud computing. InFifth International Conference on Computing, Communications and Networking Technologies (ICCCNT) 2014 Jul 11 (pp. 1-5). IEEE.

[13] Sajjan RS, Yashwantrao BR. Load balancing and its algorithms in cloud computing: a survey. International Journal of Computer Sciences and Engineering. 2017 Jan;5(1):95-100.

[14] Malhotra M. A review, different improvised throttled load balancing algorithms in cloud computing environment". Int. J. Eng. Technol. Manag. Appl. Sci.. 2017;5(7):410-6.

[15] Megharaj G, Mohan KG. A survey on load balancing techniques in cloud computing. IOSR Journal of Computer Engineering (IOSR-JCE). 2016 Apr;18(2):55-61.

[16] Sachdeva R, Kakkar S. A novel approach in cloud computing for load balancing using composite algorithms. Int J. 2017 Feb;7(2):198.

[17] Rathore J, Keswani B, Rathore VS. An efficient load balancing algorithm for cloud environment. J Invent Comput Sci Commun Technol. 2018;4(1):37-41.

[18] Khanchi M, Tyagi S. An efficient algorithm for load balancing in cloud computing. International Journal of Engineering Sciences & Research Technology. 2016 Jun;5(6):468-75.

[19] Babu KR, Joy AA. Samuel (2017) Load balancing of tasks using hybrid technique with analytical method of esce & throttled algorithm. Int J Nov Res Dev.;2(6):61-6.
[20] Gopinath PG, Vasudevan SK. An in-depth analysis and study of Load balancing techniques in the cloud computing environment. Procedia Computer Science. 2015 Jan 1;50:427-32.
[21] George Amalarethinam DI, Kavitha S. Rescheduling enhanced Min-Min (REMM) algorithm for meta-task scheduling in cloud computing. InInternational Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018 2019 (pp. 895-902). Springer International Publishing.
[22] Wang, S. C., Yan, K. Q., Liao, W. P., & Wang, S. S. (2010, July). Towards a load balancing in a three-level cloud computing network. In Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on (Vol. 1, pp. 108-113). IEEE.
[23] Arshad Ali, S., Member, S., Alam, M., 2019. Resource Aware Min-Min (RAMM) Algorithm for Resource Allocation in Cloud Computing Environment, 3, pp. 1863–1870 doi: 10.35940/ijrte.C5197.098319.
[24] Patel, G., Mehta, R., Bhoi, U., 2015. Enhanced Load Balanced Min-min Algorithm for Static Meta Task Scheduling in Cloud Computing. Procedia Comput. Sci. 57, 545–553.
[25] Shanthan BH, Arockiam L. Resource based load balanced min min algorithm (RBLMM) for static meta task scheduling in cloud. InInternational conference on advances in computer science and technology. Int J Eng Technol Spec 2018 (No. 1-8).
[26] Villanueva, J.C., 2015. Comparing Load Balancing Algorithms," 28-Jun-2015. [Online]. Available: https://www.jscape.com/blog/load-balancing-algorithms. [Accessed: 18-May-2020].
[27] Tailong V, Dimri V. Load balancing in cloud computing using modified optimize response time. International Journal of Advanced Research in Computer Science and Software Engineering. 2016 May;6(5).
[28] Kaurav NS, Yadav P. A genetic algorithm-based load balancing approach for resource optimization for cloud computing environment. Int J Inf Comput Sci. 2019;6(3):175-84.
[29] Richhariya V, Dubey R, Siddiqui R. Hybrid technique for load balancing in cloud computing using modified round robin algorithms. J Comput Math Sci. 2015 Dec;6(12):688-95.
[30] Kiritbhai PB, Shah NY. Optimizing load balancing technique for efficient load balancing. Int J Innov Res Technol. 2017 Nov;4(6):39-44.
[31] Ehsanimoghadam P, Effatparvar M. Load balancing based on bee colony algorithm with partitioning of public clouds. International Journal of Advanced Computer Science and Applications. 2018;9(4).
[32] Gandhi, R., Genetic Algorithms - Data Driven Investor - Medium," 12-May-2018. [Online]. Available: https://medium.com/datadriveninvestor/geneticalgorithms- 9f920939f7cc. [Accessed: 12-Jun-2020].
[33] Wang T, Liu Z, Chen Y, Xu Y, Dai X. Load balancing task scheduling based on genetic algorithm in cloud computing. In2014 IEEE 12th international conference on dependable, autonomic and secure computing 2014 Aug 24 (pp. 146-152). IEEE.
[34] Pilavare MS, Desai A. A novel approach towards improving performance of load balancing using genetic algorithm in cloud computing. In2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS) 2015 Mar 19 (pp. 1-4). IEEE.
[35] Khajemohammadi H, Fanian A, Gulliver TA. Fast workflow scheduling for grid computing based on a multi-objective genetic algorithm. In2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM) 2013 Aug 27 (pp. 96-101). IEEE.
[36] Parmesivan YA, Hasan S, Muhammed A. Performance evaluation of load balancing algorithm for virtual machine in data centre in cloud computing. Int. J. Eng. Technol. 2018;7(4.31):386-90.
[37] Liu Z, Wang X. A PSO-based algorithm for load balancing in virtual machines of cloud computing environment. InAdvances in Swarm Intelligence: Third International Conference, ICSI 2012, Shenzhen, China, June 17-20, 2012 Proceedings, Part I 3 2012 (pp. 142-147). Springer Berlin Heidelberg.
[38] Zhan S, Huo H. Improved PSO-based task scheduling algorithm in cloud computing. Journal of Information & Computational Science. 2012 Nov 1;9(13):3821-9.
[39] Pathan AF, Mallikarjuna SB. A load balancing model based on cloud partitioning for the public cloud. International journal of information and computation technology. 2014;4(16).
[40] Bachir B, Ali A, Abdellah M. Multiobjective optimization of an operational amplifier by the ant colony optimisation algorithm. Electrical and Electronic Engineering. 2012;2(4):230-5.
[41] Mishra R, Jaiswal A. Ant colony optimization: A solution of load balancing in cloud. International

Journal of Web & Semantic Technology. 2012 Apr 1;3(2):33-50.

[42] Chen WN, Zhang J. An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). 2008 Oct 31;39(1):29-43.

[43] Gao Y, Guan H, Qi Z, Hou Y, Liu L. A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. Journal of computer and system sciences. 2013 Dec 1;79(8):1230-42.

[44] https://www.cloudsimtutorials.online/cloudsim-simulation-toolkit-an-introduction.

COPYRIGHTS
©2023 by the authors. Published by the Islamic Azad University, Khodabandeh Branch, Zanjan. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 International (CC BY 4.0) https://creativecommons.org/licenses/by/4.0