International Conference on
RESEARCH IN ELECTRICAL AND COMPUTER ENGINEERING
کنفرانس بین المللی پژوهش در مهندسی برق و کامپیوتر
15 December 2016 Singapore www.receconf.com سنگاپور ۲۵ آذر ۱۳۹۵

ارائه یک الگوریتم تعادل بارترکیبی برای سرویس های محاسبات ابری

پری عظیمی صفت
دانشجوی کارشناسی ارشد گروه کامپیوتر ، واحد سنندج ، دانشگاه آزاد اسلامی، سنندج، ایران
azimi.pari90@gmail.com
کیهان خامفروش
استادیار و عضو هیأت علمی گروه کامپیوتر ، واحد سنندج ، دانشگاه آزاد اسلامی ، سنندج ، ایران
khamfroosh@yahoo.com

چکیده
محاسبات ابری یک الگوی محاسباتی است که به اشتراک داده ها، محاسبات و شفافیت سرویس روی یک شبکه‌ی مقیاس‌پذیر از گره‌ها کمک می‌کند. یکی از مؤلفه های مهم یک معماری رایانش ابری، مؤلفه ی توازن بار است. مهم ترین وظیفه ی این مؤلفه، دریافت کارهای کاربران و توزیع آنها بر روی سرورها و میزبان های متفاوت می باشد، به گونه ای که درصد به کارگیری منابع سرورها تقریباً یکسان و محاسبات نیز به صورت متوازن بر روی هر سرور قرار گیرند. این عمل منجر به افزایش رضایت کاربران و افزایش میزان بهره وری از منابع خواهد شد. در این مقاله با استفاده از روش جستجوی هارمونی و الگوریتم ژنتیک به ارائه یک روش ترکیبی جدید تعادل بار برای سیستم های ابری پرداخته ایم. در ادامه روش پیشنهادی را مورد ارزیابی قرار داده ایم. معیار های ارزیابی کاهش زمان Makespan و افزایش میزان speedup می باشد. نتایج شبیه سازی نشان می دهد که روش پیشنهادی به مراتب نتایج بهتری را نسبت به الگوریتم های مورد مقایسه تولید می کند.

واژگان کلیدی : محاسبات ابری،تعادل بار، الگوریتم جستجوی هارمونی (HS)، زمان Makespan ، میزان تسریع .

مقدمه

محاسبات ابری(Cloud Computing) یک الگوی محاسباتی است که به اشتراک داده ها، محاسبات و شفافیت سرویس روی یک شبکه‌ی مقیاس‌پذیر از گره‌ها کمک می‌کند. ابر شامل بسیاری از منابع سخت افزاری و نرم افزاری و مدیریت آن نقش مهمی در اجرای درخواست های مشتریان بازی می کند. از آنجا که محاسبات ابری، داده‌ها را در یک محیط باز ذخیره می‌کند، بنابراین میزان داده‌های ذخیره شده به سرعت افزایش می‌یابد. در ذخیره‌سازی ابری، تعادل بار(Load Balancing) یک مساله کلیدی است. تعادل بار یکی از چالش‌های اصلی در محاسبات ابری است و یک روش برای توزیع پویای بارکاری (work load) روی چند کامپیوتر درشبکه می باشد، که موجب استفاده بهینه از منابع، ماکزیمم کردن توان عملیاتی (throughput) و کاهش زمان پاسخ است . با استفاده از تعادل بار، کارایی کلی سیستم بهبود یافته و همچنین تضمین می‌کند که منابع محاسباتی به طور کارآمد و منصفانه روی شبکه توزیع شوند. برای این منظ.ور الگوریتم‌های مختلفی بکار رفته است (Sidhu, Kinger,2013) معیارهایی که برای ارزیابی این الگوریتم‌ها در ابر در نظر گرفته می‌شوند، معمولا شامل مواردی نظیر مقیاس‌پذیری، میزان بهره وری از منابع، کارایی، زمان پاسخ و توان عملیاتی به معنی تعداد کارهای اجرا شده در واحد زمان می باشد(Sidhu , Kinger,2013; Desai, Prajapati , 2013)

به طور کلی الگوریتم‌های تعادل بار از دو دیدگاه قابل دسته بند ی هستند (Rajan, Jeyakrishnan,2013) . دیدگاه اول شامل الگوریتم‌های متمرکز، توزیع شده و ترکیبی می باشد. که در رویکردهای تعادل بار متمرکز یک گره مسئول مدیریت توزیع بار در تمام سیستم است. در رویکرد تعادل بار توزیع شده هر گره به طور مستقل، با جمع‌آوری اطلاعات بار سایر گره‌ها، بردار بار خودش را می‌سازد و بر اساس بردارهای بار محلی تصمیم می‌گیرد. اغلب، این نوع رویکرد برای محاسبات ابری مناسب‌تر هستند. رویکردهای ترکیبی نیز ترکیبی از دو رویکرد متمرکز و توزیع شده هستند

در دیدگاه دوم به الگوریتم های تعادل بار به الگوریتم های ایستا و پویا قابل دسته بندی هستند. الگوریتم های توازن بار ایستا به وضعیت فعلی سیستم بستگی ندارند و نیاز به دانش و آگاهی قبلی از سیستم می باشد و سازگار با تغییرات بار در زمان اجرا نیستند . هدف این نوع الگوریتم ها ،به حداقل رساندن زمان اجرای کار و محدود کردن سربار ارتباطات و تأخیر است. رویکرد تعادل بار پویا وضعیت فعلی سیستم را در طول تصمیمات تعادل بار در نظر می‌گیرد. الگوریتم‌های تعادل بار پویا، کارایی بهتری نسبت به الگوریتم‌های ایستا دارند (Ghuge, Doorwar, 2014; Divya, et al, 2014) . در این تحقیق قصد داریم که با بهره گیری از روش‌های بهنیه سازی بتوانیم روشی جدید از تعادل بار را ارائه کنیم که باعث افزایش کارایی الگوریتم در محاسبات ابری شود. در این روش با بکارگیری الگوریتم ژنتیک (Juntao, et al, 2016; Saima, et al, 2016) و با در نظر گرفتن زمان پاسخ هر درخواست یک بهبودی در توازن بار در محیط محاسباتی ایجاد خواهد شد. الگوریتم پیشنهادی ما ترکیبی از خصوصیات بهینه الگوریتم های ژنتیک (GA) و جستجوی هارمونی (HS) می باشد. در روش پیشنهادی با استفاده از الگوریتم هارمونی منابع در دسترس بدست می آیند و سپس با استفاده از کوتاه ترین مسیر به دست آمده در بین منابع در دسترس و بکارگیری الگوریتم ژنتیک بهترین منبع را به کارها (Tasks) تخصیص می دهیم.

کارهای انجام شده

روش های مختلفی برای ایجاد تعادل بار در ابر مطرح شده است که هریک دارای مزایا و معایبی هستند. در ادامه به بررسی بعضی از این الگوریتم ها خواهیم پرداخت. در (Pandey, Wu, Guru, Buyya, 2015) یک الگوریتم کلونی مورچه تطبیقی برای زمان بندی منابع معرفی شده است. این الگوریتم بر مبنای اجرای یک سرویس ابر بر روی شبکه ای با پهنای باند محدود و با

تنظیم یک آستانه مناسب و نزدیک به ظرفیت حداقلی شبکه برای جستجوی کوتاه ترین مسیر استفاده شده کار می کند. در این روش ، کوتاهترین مسیر انتخاب می شود و در صورت ایجاد ازدحام و تصادف در آن مسیر با توجه به مقدار آستانه، مسیر نیمه بهینه دیگری انتخاب شده و الگوریتم ادامه می یابد. در روش ارائه شده توسط (Chen, 2014) از الگوریتم PSO به منظور یافتن بهترین نگاشت منبع استفاده شده است و نتایج حاصل از اجرای آن با روش BFS مقایسه می گردد. الگوریتم LAGA توسط نویسندگان (لرکی محمدی و ساجدی،۱۳۹۲) به منظور سیستم های گسترش یافته گرید و کلود طراحی شده است که بر اساس الگوریتم ژنتیک کار می کند. LAGAیک روش ترکیبی مبتنی بر الگوریتم ژنتیک و آتوماتای یادگیر می باشد. الگوریتم NGA در (Jafari, Alty, Chambers, 2014)برای سیستم های چندپردازنده ی ناهمگن طراحی و ارائه شده است. این الگوریتم نیز بر اساس الگوریتم ژنتیک کار می کند. نویسندگان مقاله (Sran, Kaur, 2013) الگوریتمی با مجموعه ای از وظایف(tasks)تخصیص داده نشده،ارائه نمودند. ابتدا زمان اتمام حداقل برای تمام وظایف محاسبه می شود. سپس در میان این زمان های حداقل، کمترین مقدار انتخاب می شود .سپس مطابق با زمان حداقل، وظیفه روی ماشین مربوطه زمانبندی می شود. این روش یک اشکال اصلی دارد که می تواند منجر به گرسنگی شود. در این مرجع تقریباً الگوریتم Min-Max ارائه شده است.

الگوریتم تعادل بار زنبور عسل در مرجع (Dhinesh, Krishna,2013) توسط این نویسندگان معرفی شده است. این الگوریتم بر مبنای رفتار زنبور عسل برای جست‌وجوی غذا مدل‌سازی شده است. این روش اولویت کارهای منتظر در صف اجرا، در ماشین مجازی (virtual machine) را درنظر می‌گیرد. در این روش ماشین‌های مجازی گروه‌بندی شده، براساس بار روی ماشین مجازی، وظیفه روی ماشین مجازی زمانبندی می‌شوند. بعد از اجرای کار، برای جلوگیری از سربار وظیفه حذف می‌شود. در این روش دو کلاس زنبور وجود دارد: زنبور نگهبانپیش آهنگ) و زنبوری که به دنبال غذا می‌رود کاوشگر). ابتدا زنبور نگهبان به دنبال غذا می‌رود بعد از یافتن منبع غذا باز می‌گردد و به زنبور کاوشگر کارگر) خبر می‌دهد. سپس زنبوران کاوشگر به دنبال زنبوران نگهبان می‌روند تا جای غذا را مکان‌یابی کنند و شروع به جمع‌آوری آن کنند سپس دوباره به کندو باز می‌گردند. یک مشکل این الگوریتم این است که ممکن است صف وظایفی با اولویت پایین در صورت وجود صف وظایف با اولویت بالا دچار قحطی‌زدگی شوند.

الگوریتم نوبت گردشی بهبود یافته (RR) در سال ۲۰۱۲ توسط (Ahmed, Singh ,2012) ارائه شده است. این الگوریتم به دلیل عدالت و عدم قحطی‌زدگی فرایندها، بسیار رایج، و بر مبنای کوانتوم زمانی کار می‌کند. از آنجا که کوانتوم زمانی ایستا است، منجر به تعویض متن کمتر برای کوانتوم زمانی بالا و تعویض متن بیشتر در کوانتوم زمانی پایین می‌شود. افزایش تعویض متن منجر به میانگین زمان انتظار و زمان بازگشت بالا می‌شود که سربار را افزایش و کارایی سیستم را کاهش می‌دهد. بنابراین کارایی سیستم وابسته به انتخاب یک کوانتوم زمانی بهینه است . ما نیز در این مقاله به ارائه یک الگوریتم تعادل بار توزیع شده ی کارا برای سیستم های ابری پرداخته ایم که در بخش بعدی به ارائه جزئیات بیشتری از الگوریتم پیشنهادی می پردازیم.

الگوریتم پیشنهادی

در این بخش به ارائه روش پیشنهادی توازن بار می پردازیم که ترکیبی از الگوریتم های ژنتیک (GA) و جستجوی هارمونی (HS) می باشد .

همانطور که قبلا نیز گفته شد، یکی از مؤلفه های مهم یک معماری رایانش ابری، مؤلفه ی توازن بار است. مهم ترین وظیفه ی این مؤلفه، دریافت کارهای کاربران و توزیع آنها بر روی سرورها و میزبان های متفاوت می باشد، به گونه ای که دسترسی به داده ها تا حد امکان محلی شده و محاسبات نیز به صورت متوازن بر روی هر سرور قرار گیرند. این عمل رضایت کاربران را حداکثر، زمان پاسخ را حداقل ، بهره برداری از منابع را افزایش، و کارایی سیستم را بالا می برد .هدف اصلی توازن بار، بهینه

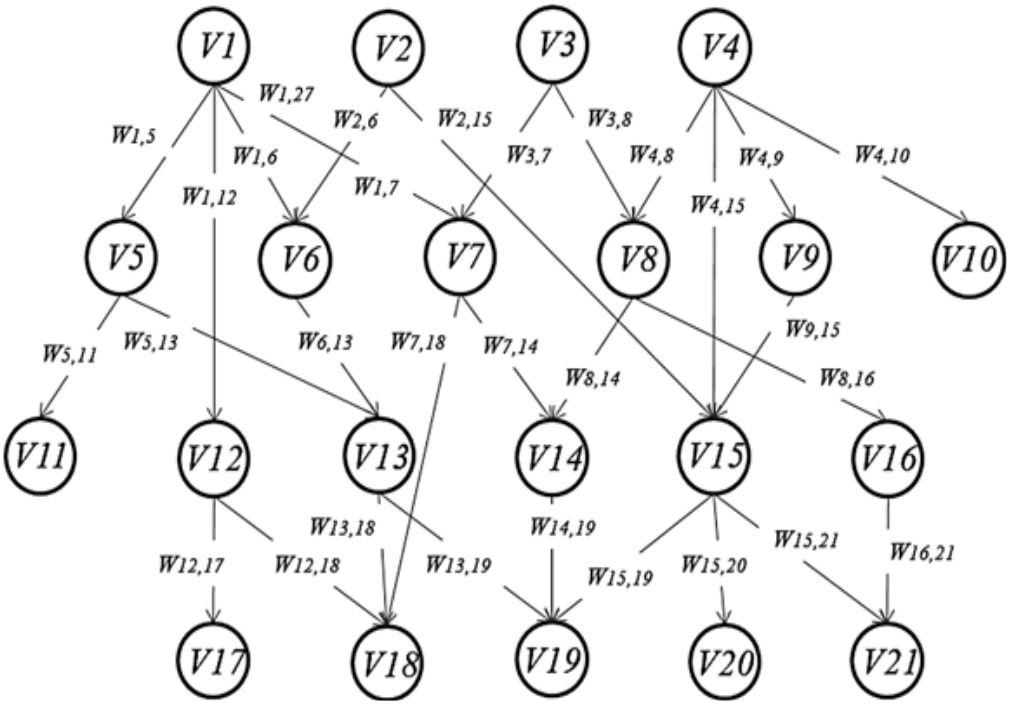
سازی کارایی و توان عملیاتی است، به طوری که زمان پاسخ نیز کاهش پیدا کند(Divya, et al,2014) . در ادامه ابتدا به ارائه چند تعریف می پردازیم.

تعریف زمان Makespan: زمان اتمام آخرین کار روی آخرین پردازنده در سیستم می باشد. این زمان برای اندازه گیری سودمندی سیستم بسیار مهم است. هر چه قدر Makespan کمتر باشد کارایی الگوریتم بهتر است (Saima, et al, 2016)

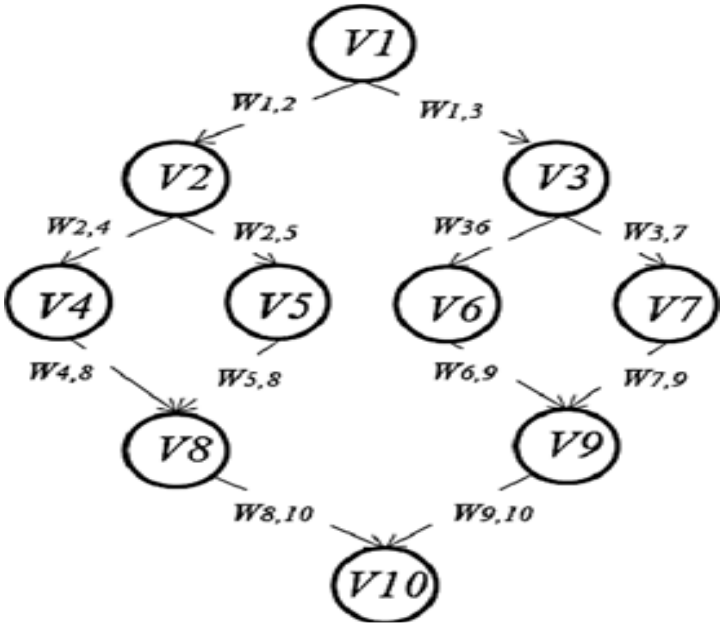
تعریف بار کاری: بار کاری همان ورودی سیستم است. بار کاری روی هر یک از ماشین های مجازی می تواند عملکرد سرور اصلی و سایر ماشین های روی آن را تحت تاثیر قرار دهد.

تعریف Speedup : تسریع(Speedup) از تقسیم زمان اجرای ترتیبی بر زمان اجرای موازی Makespan محاسبه می شود. به عبارتی این پارامتر نشان دهنده میزان سرعت پایان اجرای کارها(وظایف) می باشد. افزایش فاکتور تسریع در یک الگوریتم به عنوان یک پارامتر بهبود کارایی سیستم شناخته شده است.

در این پژوهش برای حل مسئله توازن بار از ترکیب دو الگوریتم ژنتیک و هارمونی استفاده شده است. الگوریتم ژنتیک با وجود توانایی های بالا در جستجوی فضای مسئله، از نظر پایداری و جستجوی محلی ضعیف عمل می کند. هدف از ارائه الگوریتم پیشنهادی ترکیب قابلیت های جستجوی عمومی الگوریتم ژنتیک با جستجوی محلی الگوریتم هارمونی است. اما در حالت های که منابع در دسترس وجود ندارد و مکانیزمی برای شناسایی این منابع در دسترس نیست ابتدا باید به یک الگوریتم مشخص برای به دست آوردن منابع بپردازیم و سپس بتوانیم براساس الگوریتم ژنتیک این منابع را به کارها تخصیص دهیم که بتوان به یک محیط با توازن بار رسید.



شکل ۱: یک گراف با ساختار پیچیده



شکل ۲: یک گراف با ساختار ساده

بنابراین برای ایجاد مکانیزم توازن در بین داده ها و منابع باید ابتدا منابع در دسترس را شناسایی و از مکان قرار گیری هریک اطلاع پیدا کرد. به همین منظور ابتدا می توان با استفاده از الگوریتم جستجوی هارمونی که یک الگوریتم ابتکاری برای به دست آوردن کوتاه ترین مسیر در یک گراف است یک گراف پیچیده (شکل یک) از منابع در دسترس را به یک گراف ساده (شکل دو) تبدیل کنیم. سپس با استفاده از الگوریتم ژنتیک بتوان این منابع را به نحوی به کارها تخصیص دهیم که بتوان به ازای بهترین تابع برازندگی سیستم فعالیت کنیم. باید به این نکته توجه داشت که در محیط های پیچیده استفاد از گراف های پیچیده (شکل یک) در حل این مسئله می تواند بسیار مشکل ساز بوده و منابع در دسترس را برای ما دور کند. به همین منظور تلاش می شود که ابتدا با استفاد از الگوریتم های دیگر مانند الگوریتم هارمونی منابع در دسترس را به دست آورد سپس با استفاده از کوتاه ترین مسیر به دست آمده در بین منابع در دسترس با استفاده از الگوریتم ژنتیک بهترین منبع را به کارها تخصیص دهیم تا بهترین نتیجه حاصل شود.

در رایانش ابری تعدادی مرکز داده وجود دارد و در مراکز داده، میزبان ها و سرورها هستند که روی آنها ماشین های مجازی وجود دارند. اینکه ماشین مجازی روی کدام میزبان قرار گیرد و درخواست رسیده از سمت کاربر را کدام ماشین مجازی اجرا کند، دو مسأله متفاوت است که به اولی مجازی سازی در سطح میزبان و به دومی مجازی سازی در سطح ماشین مجازی گفته می شود. درخواست هایی که از سمت کاربران به سمت مراکز داده می آیند باید به نحوی اجرا شوند که توازن بار برقرار شود. الگوریتم ها و روش های مختلفی در این زمینه وجود دارند که مزایا و معایب هر کدام از الگوریتم ها و اینکه هر کدام منجر به بهبود چه پارامتری می شوند قبلا" مورد بحث قرار است . در ادامه به جزئیات الگوریتم پیشنهاد شده می پردازیم. در الگوریتم پیشنهادی به این صورت عمل می شود که درخواست ها باید به نحوی به ماشین های مجازی اختصاص داده شوند که زمان پاسخ و Makespan بهینه شوند . اگر درخواستی برای اجرا وجود داشته باشد آن درخواست به ماشینی اختصاص داده شود که بارکاری روی آن کمتر و قدرت پردازشی آن، بالاتر باشد. دلیل در نظر گرفتن این دو پارامتر به طور همزمان، این است که اگر انتخاب فقط براساس بارکاری روی ماشین مجازی صورت گیرد و ماشینی با بارکاری کم، انتخاب شود، ممکن است درخواست جاری نیاز به ماشین مجازی با قدرت پردازش بالا داشته باشد اما اگر قدرت پردازش ماشین مجازی انتخاب شده پایین باشد، در این صورت انتخاب درستی صورت نگرفته است. اگر انتخاب فقط براساس قدرت پردازش بالای ماشین مجازی باشد، ممکن است ماشینی با قدرت پردازش بالا انتخاب شود ولی بارکاری روی آن به قدری زیاد باشد که درخواست مجبور

شود مدت زمان زیادی را صبر کند، بنابراین باید به هر دو پارامتر به طور همزمان توجه شود و ماشینی انتخاب شود که بارکاری روی آن کم و قدرت پردازش آن بالا باشد. با توجه به بارکاری روی ماشین مجازی، زمان انتظار برای درخواست جاری و با توجه به سرعت پردازنده ،زمان پردازش برای درخواست جاری به دست می آید. در نهایت، درخواست به ماشینی اختصاص داده می شود که مجموع زمان پردازش و انتظار آن ،کمترین مقدار را داشته باشد.

هر درخواست یا کار، یک طول ($Length\ Cloudlet$) دارد، که به معنی تعداد دستورالعمل های درونی آن می باشد. زمان انتظار به صورت زیر محاسبه می شود:

$$\text{Waiting Time} = \frac{length\ (cloudlets\ on\ VM)}{MIPS\ (CPUVM)} \qquad (۱)$$

$length$ cloudlets $on\ VM$: تعداد دستورالعمل های درخواست هایی که روی ماشین مجازی قرار دارند را نشان می دهد، که همان بار روی ماشین مجازی می باشد.

$length\ \ cloudlet\ current$: طول یا تعداد دستورالعمل های درخواست جاری را نشان می دهد.

زمان انتظار، ارتباط مستقیمی با بار روی ماشین مجازی دارد. به این معنی که هر چه قدر بار روی ماشین مجازی بیشتر باشد، زمان انتظار بیشتر و هر چه بار روی ماشین مجازی کمتر باشد، زمان انتظار کمتر می شود.

$MIPS\ CPUVM$: MIPSسرعت پردازنده ی ماشین مجازی را نشان می دهد که واحد آن میلیون دستورالعمل در واحد ثانیه می باشد. هر چه قدرت پردازش ماشین مجازی بالاتر باشد زمان اجرا کمتر و هر چه قدرت ماشین مجازی پایین تر باشد، زمان اجرا طولانی تر می شود. بنابراین مراحل الگوریتم پیشنهادی به صورت ذیل می باشد:

مفروضات الگوریتم: فرض می شود لیستی از درخواست ها (کارها) و ماشین های مجازی موجود هستند.

شروع الگوریتم

گام ۱: برای تعیین اینکه درخواست مورد نظر به کدام ماشین مجازی اختصاص داده شود، مراحل ذکر شده در پایین برای هر درخواست انجام شود.

گام ۲: کنترل کننده مرکز داده درخواست جدیدی را دریافت می کند.

گام ۳: وقتی درخواست جدیدی از سوی کاربر می آید کنترل کننده مرکز داده به متوازن کننده بار، گزارش می دهد تا متوازن کننده ی بار، ماشین مجازی مناسب را برای اجرای آن درخواست، پیدا کند.

گام ۴: متوازن کننده بار، بارکاری همه ی ماشین های مجازی را به دست می آورد. برای این منظور، باید زمان انتظار را برای درخواست جاری روی همه ی ماشین های مجازی محاسبه شود.

گام ۵: متوازن کننده بار، قدرت پردازش ماشین مجازی را به دست می آورد. برای این منظور باید، زمان اجرا را برای درخواست جاری، بر روی همه ی ماشین های مجازی محاسبه کند.

این زمان از طریق فرمول زیر محاسبه می شود که با قدرت پردازش ماشین مجازی رابطه ی عکس دارد.

$$\text{Execution Time} = \frac{length\ (cloudletcurrent)}{MIPS\ (CPUVM)} \qquad (2)$$

گام ۶: با توجه به مراحل 4 و 5، متوازن کننده بار، زمان پاسخ درخواست جاری، که مجموع زمان انتظار و اجرا است را برای همه ی ماشین های مجازی محاسبه می کند .

$$\text{Response Time} = \text{Execution Time} + \text{Waiting Time} \qquad (۳)$$

گام ۷: متوازن کننده بار، ماشین مجازی با کمترین زمان پاسخ را شناسایی نماید.یعنی ماشینی که بار روی آن کمتر وقدرت پردازش آن بالاتر است .

گام ۸: متوازن کننده بار،شناسه ماشین مجازی انتخاب شده در مرحله ۷ را ، به کنترل کننده مرکز داده می فرستد .

گام ۹: کنترل کننده مرکز داده، درخواست را به ماشین مجازی انتخاب شده،اختصاص می دهد تا آن را اجرا کند.

گام ۱۰: تا زمانی که در خواست جدیدی وجود دارد مراحل ۲ تا ۹ را برای این درخواست ها اجرا کن.

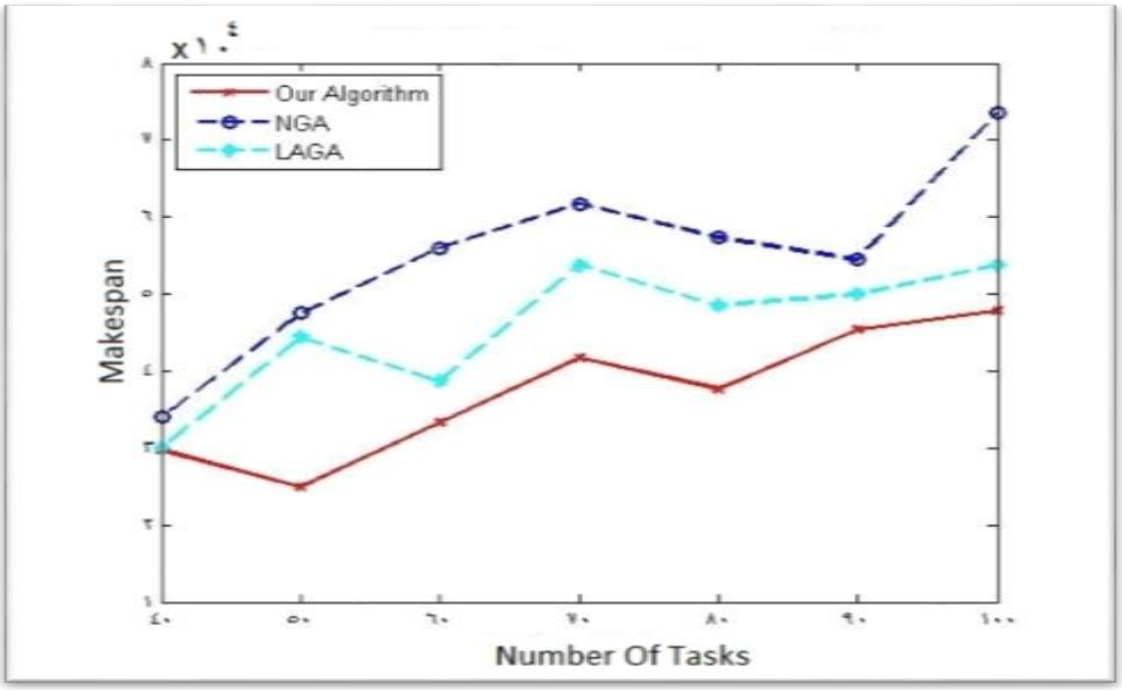
پایان الگوریتم.

نتایج شبیه سازی

در این بخش به شبیه سازی و ارزیابی الگوریتم پیشنهادی خود می پردازیم. به منظور ارزیابی از نرم افزار شبیه ساز MATLAB استفاده نموده ایم. پارامترهای ارزیابی را Makespan یعنی زمان اتمام آخرین کار در سیستم و زمان Speedup در نظر گرفته ایم. زیرا کاهش زمانMakespan و افزایش میزان speedup از مهم ترین فاکتورهای کیفیت سرویس سیستم های ابری می باشند.

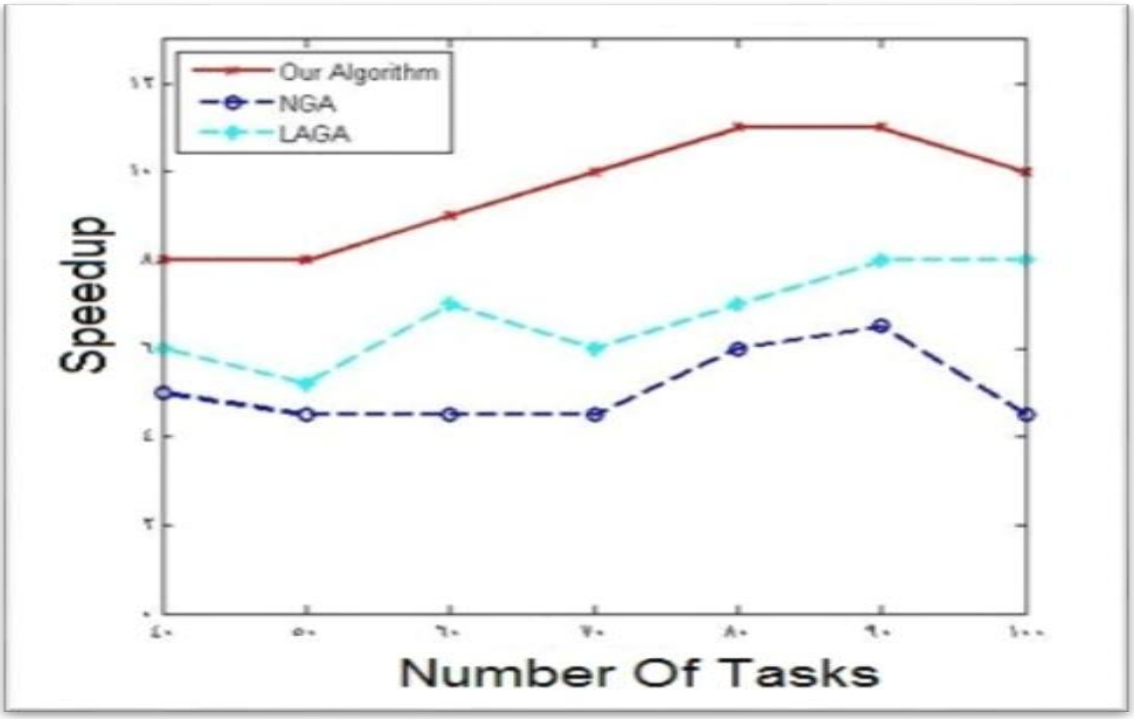
فضای آزمایش را یک سیستم محاسبات ابری با تعداد ۱۰ پردازنده با توان های پردازشی متفاوت در نظر می گیریم که بصورت تصادفی در محیط توزیع شده اند. به عبارتی هر پردازنده دارای اندازه ی MIPS مخصوص به خود است که می تواند با پردازنده ی دیگر مقدارش متفاوت باشد. همچنین تعداد کارها درخواست ها) را بصورت متغیر و از تعداد ۴۰ تا ۱۰۰ کار در نظر گرفته ایم، که مدت زمان اجرای هر کار نیز می تواند با کار دیگر متفاوت باشد. شبیه سازی را بر اساس الگوریتم پیشنهادی اجرا می کنیم. به منظور دستیابی به نتایج دقیق تر شبیه سازی را چندین مرتبه

اجرا کرده و میانگین نتایج بدست آمده از اجرای شبیه سازی را محاسبه می کنیم. نمودار شکل نتایج ارزیابی و مقایسه الگوریتم پیشنهادی را با دو الگوریتم LAGA و NGA از نظر پارامتر Makespan همزمان با افزایش تعداد کارها را نشان می دهد. همان طور که در شکل ۳ نشان داده شده است، الگوریتم پیشنهادی با تعداد درخواست های مختلف، در مقایسه با الگوریتم های LAGA و NGA عملکرد بهتری داشته است و زمان Makespan کمتری دارد.



شکل۳: نمودار مقایسه Makespan با توجه به افزایش تعداد وظایف

نمودار شکل ۴ نیز به بررسی مقایسه ارزیابی روش پیشنهادی با این دو الگوریتم از نظر پارامتر Speedup ضمن افزایش تعداد کارهاوظایف) پرداخته است.



شکل۴: نمودار مقایسه Speedup الگوریتم به ازای افزایش تعداد وظایف

همانطور که قبلا اشاره شد، افزایش فاکتور تسریع سبب بهبود کارایی سیستم می شود. با توجه به نمودار شکل ۴ میزان تسریع Speedup) روش پیشنهادی در مقایسه با الگوریتم های مورد مقایسه بیشتر است و روش پیشنهاد شده به مراتب نتایج بهتری را از نظر این پارامتر نیز تولید نموده است.

نتیجه گیری

تعادل بار یکی از مسائل مهم و حائز اهمیت در سرویس های محاسبات ابری و یک روش برای توزیع بارکاری (work load) روی چند پردازنده می باشد، که سبب استفاده بهینه از منابع خواهد شد. در این پژوهش یک الگوریتم تعادل بار را برای حل مسئله توازن بار در سیستم محاسبات ابری پیشنهاد دادیم، که در آن از ترکیب دو الگوریتم ژنتیک و هارمونی استفاده شده است. در ادامه روش پیشنهادی را با الگوریتم های LAGA و NGA از نظر پارامترهای Makespan و Speedup مورد مقایسه قرار دادیم و نشان دادیم که روش پیشنهادی نتایج بهتری را از نظر این پارامترها نسبت به سایر الگوریتم ها تولید کرده است.

منابع

لرکی محمدی، آرش و ساجدی هدیه ۱۳۹۲)، الگوریتم LAGA یک روش ترکیبی مبتنی بر الگوریتم ژنتیک و اتوماتای یادگیرجهت بهینه سازی توابع پیچیده، دوازدهمین کنفرانس ملی سیستم های هوشمند، بم، انجمن سیستم های هوشمند ایران.

Ahmed, T. and Singh Y. 2012). Analytic Study of Load Balancing Technique Using Tool Cloud Analyst. Journal of Engineering Research and Applications IJERA). Vol. 2. 1027-1030.

Chen, Z. 2014). Adaptive Ant Colony Algorithm Researching in Cloud Computing Routing Resource Scheduling. 19th International Conference on Industrial Engineering and Engineering Management. Springer. 101.108.

Desai, T. and Prajapati J. 2013). A Survey of Various Load Balancing Techniques and Challenges in Cloud Computing. International journal of scientific and technology research.

Dhinesh, B. Krishna, L, D. 2013). Honey bee behavior inspired load balancing of tasks in cloud computing environments. In proc. Applied Soft Computing. vol. 13. Issue 5. May. 2292-2303.

Divya, V, V. Bharathi and et al. 2014). A Survey on Load Balancing in Public Cloud. International Journal of Advanced Research in Computer Science and Software Engineering.

Ghuge, K. and Doorwar M. 2014). A Survey of Various Load Balancing Techniques and Enhanced Load Balancing Approach in Cloud Computing. International Journal of Emerging Technology and Advanced Engineering.

Jafari, M, G. Alty, S, R. and Chambers, J, A. 2014). New natural gradient algorithm for cyclostationary sources. IEE Proc.-Vis. Image Signal Process. Vol. 151. No. 1. February2014.

Juntao, Ma. et al. 2016). A Novel Dynamic Task Scheduling Algorithm Based on Improved Genetic Algorithm in Cloud Computing. Wireless Communications. Networking and Applications. Springer India .829-835.

Pandey, S. Wu, L. Guru, S, M. And Buyya R. 2015). A Particle Swarm optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments. 24th IEEE International Conference on Advanced Information Networking and Applications. IEEE Computer society.

Rajan, R, G. and Jeyakrishnan V. 2013). Survey on Load Balancing in Cloud Computing Environments International Journal of Advanced Research in Computer and Communication Engineering.

Saima, A. Gulzar. et al. 2016). A hybrid genetic algorithm for optimization of scheduling workflow applications in heterogeneous computing systems. Journal of Parallel and Distributed Computing. 80-90.

Sidhu, A, K. and Kinger, S. 2013). Analysis of Load Balancing Techniques in Cloud Computing. International Journal of Computers and Technology.

Sran, N. and Kaur, N. 2013). Comparative Analysis of Existing Load Balancing Techniques in Cloud Computing. International Journal of Engineering Science Invention. Vol.2. Issue 1.