علوم رایانش و فناوری اطلاعات
نشریه علمی انجمن کامپیوتر ایران
مجلد ۱۳، شماره ۲، ۱۳۹۴
صفحات ۱۴-۲۲
مقاله عادی

ارائه روش بهبود یافته زمان‌بندی کارها در محیط ابر با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات

فاطمه عبادی‌فرد احمد اکبری

دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران

چکیده

محاسبات ابری یکی از زمینه‌هایی است، که در سال‌های اخیر توجه کاربران زیادی را به خود جلب کرده است، این امر به دلیل مزایای قابل توجهی است، که سرویس‌های ابر از لحاظ هزینه و کارایی در اختیار مصرف‌کنندگان قرار می‌دهند. مسئله ی زمان بندی کارها یکی از مهمترین مسائل جهت استفاده مناسب از قابلیت های محیط ابر می باشد. به طور کلی مسئله زمان‌بندی کارها در محیط ابر یک مسئله ان‌پی-سخت می‌باشد [۱۶]. در مسئله‌ی زمان‌بندی کارها از یک طرف باید نیازمندی‌های کاربران در نظر گرفته شود و از طرف دیگر باید از منابع موجود حداکثر استفاده شود. در این مقاله روشی را برای زمان‌بندی کارها با استفاده از رویکرد بهبود یافته‌ی الگوریتم بهینه‌سازی ازدحام ذرات[۱] به کمک استفاده از الگوریتم تعادل بار برای تولید نسل اولیه مناسب‌تر، ارائه کرده‌ایم که روش پیشنهادی علاوه بر آنکه بار کاری را بین ماشین‌های مجازی موجود متعادل کرده، با انتخاب تابع برازندگی مناسب، سبب کاهش زمان تکمیل تمامی کارها[۲]، همچنین استفاده‌ی حداکثر از منابع، می‌شود. نتایج شبیه‌سازی نشان می‌دهد که روش پیشنهادی در مقایسه با روش بهینه‌سازی ازدحام ذرات بدون استفاده از تعادل بار، به‌میزان ۳۳ درصد بهبود در زمان تکمیل تمامی کارها و به میزان ۲۲ درصد بهبود در میزان کارایی منابع دارد.

کلمات کلیدی: محاسبات ابری، زمان‌بندی، بهینه‌سازی ازدحام ذرات، تعادل بار، بهره‌وری منابع.

۱- مقدمه

امروزه محاسبات ابری یکی از تکنولوژی‌های جدیدی است که به طور کامل بر پایه اینترنت می‌باشد و در آن همه‌ی برنامه‌ها و فایل‌ها بر روی ابری قرار می‌گیرد، که شامل هزاران کامپیوتر به هم متصل می‌باشد. محاسبات ابری ترکیبی از محاسبات توزیع شده و موازی به‌منظور فراهم کردن منابع اشتراکی از قبیل سخت افزار، نرم‌افزار و اطلاعات از دیگر دستگاه‌ها می‌باشد [۱]. از دید توسعه‌دهندگان نرم‌افزار، خدماتی که از طریق ابر ارائه می‌شود؛ می‌توان به سه گروه زیرساخت به عنوان خدمات[۳]، سکو به عنوان خدمات[۴] و نرم‌افزار به عنوان خدمات[۵] طبقه‌بندی کرد.

در سطح زیرساخت به عنوان خدمات، فراهم‌کننده‌ی ابر منابع سخت‌افزاری را برای اجرای خدمات با استفاده از فناوری مجازی‌سازی در اختیار کاربران قرار می‌دهد. در سطح سکو به عنوان خدمات، لایه‌ای از نرم‌افزار را برای ایجاد سرویس‌های سطح بالاتر فراهم می‌کند. این بستر توسعه‌ای، به کاربران برای طراحی، توسعه، ارزیابی و میزبانی کردن برنامه‌های کاربردی در سطح ابر کمک می‌کند. در سطح نرم‌افزار به عنوان سرویس کاربران نرم‌افزار مورد نیاز خود را از محیط ابر درخواست می‌کنند.

در میان همه‌ی سطوح سرویس‌دهی محیط‌های ابر، سطح نرم‌افزار به عنوان سرویس در محیط ابر، توجه‌ی کاربران زیادی را به خود جلب کرده است. این سطح روش‌های سنتی استفاده از نرم‌افزار را که شامل نصب نرم‌افزار بر روی ماشینی که قصد استفاده از نرم‌افزار را دارد، با میزبانی نرم‌افزار روی سرورهای راه دور تغییر داده است و سبب شده تا محدودیت‌هایی را که مانع نصب برخی از نرم‌افزارهای حجیم و یا مانع تهیه‌ی نرم‌افزار برای کاربران بوده است را از بین ببرد.

محیط ابر بستری از سرورها را در مرکز داده فراهم می‌کند تا هنگام درخواست منابع توسط کاربران آنها را به صورت اشتراکی در اختیارشان قرار دهد. فراهم‌کنندگان سرویس‌های مختلف[۶] می‌توانند با اجاره ماشین‌های مجازی از

فراهم‌کنندگان ابر، سرویس‌های مختلف خود را به کاربران خود ارائه دهند. کاربران برنامه مورد علاقه خود را از فراهم‌کننده سرویس درخواست و آن را دریافت می‌کنند.

از آنجایی که فراهم‌کنندگان سرویس منابع خود را از فراهم‌کنندگان ابر اجاره می‌کنند، با توجه به پراکندگی جغرافیایی مختلف کاربران و ویژگی‌های خاص برنامه‌های مختلف که توسط کاربران درخواست داده می‌شود، چالش اساسی در این زمینه ارائه روشی کارا برای تخصیص منابع مورد نیاز به درخواست‌های کاربران به صورتی می‌باشد، که بتواند پارامترهای کیفیت سرویس مورد نیاز متناسب با نیازمندی‌های آن برنامه را برای کاربران به همراه داشته باشد [۲].

زمان‌بندی کارها تکنیکی است که وظایف کاربران به ماشین‌های مجازی برای اجرا اختصاص می‌دهد. از دید مشتری الگوریتم زمان‌بندی مناسب باید بتواند وظایف مورد تقاضای او را با کمترین زمان بر روی ماشین مجازی اجرا کند، از طرفی فراهم‌کننده ی سرویس نیازمند نوعی زمان‌بندی است که بتواند در عین رضایت‌مندی مشتری از منابع حداکثر استفاده را بکند. این مساله نیازمندی فراهم کننده‌ی سرویس را به انتخاب روش مناسب برای زمان‌بندی درخواست‌ها بیشتر می‌کند.

در این مقاله با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات [۱۲] به همراه الگوریتم تعادل بار، یک روش برای زمان‌بندی درخواست‌ها بر روی ماشین‌های مجازی ارائه کرده‌ایم که علاوه بر کاهش زمان تکمیل تمامی کارها، سبب افزایش بهره‌وری از منابع[۷] می‌شود.

مزیت الگوریتم ارائه شده در روش فوق نسبت به روش پایه‌ی بهینه‌سازی ازحام ذرات در این است که در روش پایه‌ی بهینه‌سازی ازحام ذرات موقعیت اولیه هر ذره به صورت تصادفی انتخاب می‌شود در صورتی که در روش ارائه شده موقعیت هر ذره در ابتدا به صورت تصادفی تعیین شده و سپس با استفاده از روش تعادل بار[۸] برای توزیع عادلانه‌ی درخواست‌ها بین ماشین‌های مجازی مختلف، جا به جا می‌شود. با استفاده از این روش هر ذره‌ی اولیه خود یک ذره‌ی مناسب می‌باشد که با جابه‌جایی آن به جواب مناسب‌تر می‌رسیم. نتایج شبیه‌سازی نشان می‌دهد که روش فوق نسبت به روش پایه‌ی بهینه‌سازی ازدحام ذرات، بهبود بهتری را در زمان تکمیل تمامی کارها و میزان بهره‌وری منابع دارد. به طور خلاصه می‌توان گفت تمرکز اصلی ما در این مقاله شامل موارد زیر می‌باشد.

۱- ارائه‌ی روشی مناسب برای زمان‌بندی درخواست‌ها با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات، با هدف کاهش زمان تکمیل تمامی کارها و افزایش بهره‌وری منابع

۲- استفاده از الگوریتم تعادل بار به همراه الگوریتم بهینه‌سازی ازدحام ذرات برای توزیع عادلانه‌ی درخواست‌ها

۳- تحلیل و ارزیابی هدفمند برای نشان دادن ترکیب موثر الگوریتم بهینه‌سازی ازدحام ذرات با الگوریتم تعادل بار برای بهبود زمان تکمیل تمامی کارها و افزایش بهره‌وری منابع

باقی مقاله از بخش‌های زیر تشکیل شده است: در بخش ۲ مروری بر کارهای مرتبط داریم در بخش ۳ پس از معرفی الگوریتم بهینه‌سازی ازدحام ذرات به بیان مدل ریاضی الگوریتم پرداخته سپس جزئیات روش پیشنهادی به تفصیل بیان می‌شود و در بخش چهارم به شبیه‌سازی و ارزیابی روش پیشنهادی می‌پردازیم و در نهایت در بخش پنجم نتیجه‌گیری و کارهای آینده بیان می‌شود.

۲- کارهای مرتبط

زمان‌بندی در پردازش ابری به معنی تخصیص بهینه‌ای از درخواست‌ها به منابع محاسباتی موجود در مراکز داده می‌باشد. در زمان‌بندی، کارها را به انواع متفاوتی از منابع با توجه به محدودیت‌های مشخص شده، نسبت می‌دهیم. محدودیت‌ها به دو دسته تقسیم‌بندی می‌شود: محدودیت‌هایی که می‌تواند از طرف کاربر باشد و شامل محدودیت در اتمام کار در زمان مشخص، تعیین سررسید (مثلاً مهلت زمانی) خاص و اختصاص هزینه محدود به کار است. دسته دوم محدودیت‌ها مربوط به فراهم‌کنندگان سرویس‌های ابری می‌باشد که شامل پاسخگویی به کار کاربران در زمان اندک و بیشینه شدن سود حاصل از انجام کار و یا افزایش بهره وری منابع می‌باشد. توجه به این محدودیت‌ها بسیار لازم و ضروری است.

در رابطه با زمان‌بندی کارهای متفاوتی انجام شده است که ما آنها را در دسته‌بندی زیر ارائه می‌کنیم: الف: روش‌هایی که برای زمان‌بندی کارها از الگوریتم‌های اکتشافی استفاده کرده است. در این روش‌ها از الگوریتم‌های اکتشافی مختلف مانند ژنتیک، بهینه‌سازی ازدحام ذرات، کلونی مورچگان و سایر الگوریتم‌های اکتشافی [۴-۶] برای حل مسئله زمان‌بندی استفاده شده است. از آنجایی که پایه کار ما بر روی الگوریتم بهینه‌سازی ازدحام ذرات می‌باشد، در ادامه برخی از مهمترین روش‌هایی که از این الگوریتم استفاده شده است را بیان می‌کنیم:

پاندی و دیگر همکاران [۹] یک روش زمان‌بندی برای جریان گردش کار بر پایه الگوریتم بهینه‌سازی ازدحام ذرات ارائه کرده است. در روش پیشنهادی هدف کاهش هزینه‌ی منابع و هزینه‌ی انتقال داده می‌باشد و برای رسیدن به این هدف در ابتدا براساس متوسط هزینه محاسباتی تمامی کارها بر روی تمامی منابع موجود و متوسط هزینه انتقال داده برای هر منبع و هر کار یک وزن اختصاص می‌دهد و سپس براساس وزن اختصاص یافته، تابع هدف را برای الگوریتم بهینه‌سازی ازدحام ذرات تعیین می‌کند و الگوریتم را اجرا می‌کند. در ادامه روش ارائه شده را با روش انتخاب بهترین منبع[۹] مقایسه کرده و نتایج شبیه‌سازی نشان می‌دهد که الگوریتم ارائه شده نسبت به روش انتخاب بهترین منبع کاهش هزینه بهتری را در بر دارد. الگوریتم پیشنهادی اگرچه سبب کاهش هزینه شده است، اما برخی از معیارهای کیفیت سرویس مانند زمان انجام کار که سبب رضایت‌مندی مشتری می‌شود را در نظر نگرفته است. در سال ۲۰۰۸ لی زنگ و دیگر همکاران [۸] با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات الگوریتمی را برای زمان‌بندی بهینه‌ی کارها در محیط گرید ارائه دادند که هدف در آن کاهش زمان اتمام کار بود. تابع هدف در الگوریتم پیشنهادی کمترین زمان اجرای کار بود و در نهایت روش پیشنهادی را در شرایط مشابه با الگوریتم ژنتیک مقایسه کرده است .سپس براساس مقایسات نشان داده است که کارایی الگوریتم بهینه‌سازی ازدحام ذرات از ژنتیک بیشتر است. این مقاله اگرچه معیارهای کیفیت سرویس کمی مورد توجه قرار داده است و تابع هدف مناسبی را انتخاب نکرده است ولی با ایجاد شرایط مشابه کارایی دو الگوریتم اکتشافی بهینه‌سازی ازدحام ذرات و ژنتیک را مقایسه کرده و برتری الگوریتم بهینه‌سازی ازدحام ذرات را نشان داده است.

گیو و دیگر همکاران [۷] در سال ۲۰۱۲ روشی را برای زمان‌بندی کارها در محیط ابر بر مبنای الگوریتم بهینه‌سازی ازدحام ذرات به‌منظور کاهش زمان تکمیل کار ارائه کردند، تابع هدف در روش پیشنهادی زمان پردازش به همراه زمان انتقال داده می‌باشد. در نتیجه در روش ارائه شده هر دو مقدار زمان پردازش کار و زمان انتقال داده با هم در نظر گرفته می‌شد. روش ارائه شده باعث کاهش زمان تکمیل کار می‌شود ولی تابع هدف آن مناسب نبوده و برخی از معیارهای کیفیت سرویس متناسب را در نظر نمی‌گیرد.

همانطور که بیان شد الگوریتم بهینه‌سازی ازدحام ذرات الگوریتم مناسبی برای زمانبندی درخواست‌ها می‌باشد که می‌توانند با استفاده از تابع هدف مناسب نیازمندی‌های کاربران و فراهم‌کنندگان سرویس را برآورده کند. برخی از روش‌های ارائه شده سعی در بهبود الگوریتم بهینه‌سازی ازدحام ذرات گرفته و برای این کار این الگوریتم را با یک الگوریتم دیگر ترکیب کرده‌اند. در ادامه در دسته دوم

روش‌هایی که از ترکیب الگوریتم بهینه‌سازی ازدحام ذرات با سایر الگوریتم‌ها برای حل مسئله زمان‌بندی استفاده کرده‌اند، را بیان می‌کنیم.

در سال ۲۰۱۲، زن [۱۱] یک روش بهبود یافته‌ی بهینه‌سازی ازدحام ذرات را با هدف کاهش متوسط زمان اجرا و افزایش سرعت همگرایی ارائه داد. ایده‌ی پیشنهادی این الگوریتم ترکیب الگوریتم سرد و گرم فلزات (SA) به همراه الگوریتم بهینه‌سازی ازدحام ذرات، می‌باشد. الگوریتم سرد و گرم فلزات به صورت محلی جستجو را انجام می‌دهد، ترکیب یک الگوریتم اکتشافی مبتنی بر جمعیت مثل بهینه‌سازی ازدحام ذرات، با الگوریتم جستجوی محلی همانند الگوریتم سرد و گرم فلزات سبب همگرایی سریع‌تر برای رسیدن به جواب بهینه می‌شود. روش پیشنهادی با سایر الگوریتم‌های ژنتیک، سرد و گرم فلزات و کلونی مورچگان مقایسه شده و کارایی آن از بقیه بیشتر است. در روش پیشنهادی فقط پارامتر زمان اجرا بررسی شده و دیگر معیارهای کیفیت سرویس در نظر گرفته نشده است علاوه بر این استفاده از دو الگوریتم اکتشافی سبب افزایش پیچیدگی در الگوریتم ارائه شده می‌شود.

در سال ۲۰۱۴، عبدی [۱۴] و دیگر همکارانش یک روش زمان‌بندی با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات با هدف کاهش زمان تکمیل تمامی کارها ارائه کردند که در روش پیشنهادی از الگوریتم اختصاص کوتاه‌ترین کار به سریع‌ترین پردازنده، برای بهبود الگوریتم بهینه‌سازی ازدحام ذرات استفاده شده است. در این روش به جای تولید تصادفی نسل اولیه در الگوریتم بهینه‌سازی ازدحام ذرات، با قرار دادن وظایف کوتاه‌تر بر روی پردازنده سریعتر، نسل اولیه را مناسب‌تر انتخاب کردند و سپس با استفاده از جابه‌جایی این ذرات جمعیت و ارزیابی آن با تابع هدف جمعیت جدید را تولید می‌کند. در نهایت روش ارائه شده را با الگوریتم ژنتیک و بهینه‌سازی ازدحام ذرات، مقایسه کرده‌اند. همچنین در سال ۲۰۱۴، کوار [۱۳] و دیگر همکاران روشی را با هدف افزایش سود حاصل از منابع و بهره‌وری از منابع به کمک ترکیب الگوریتم سرد و گرم فلزات و بهینه‌سازی ازدحام ذرات، ارائه کردند.

از آنجایی که در الگوریتم پیشنهادی از ترکیب الگوریتم بهینه‌سازی ازدحام ذرات و تعادل بار استفاده شده است، مروری بر روش‌های تعادل بار انجام داده‌ایم: در رابطه با تعادل بار کارهای متفاوتی انجام شده است، به طور کلی الگوریتم‌های تعادل بار به دو دسته ایستا[۱۰] و پویا[۱۱] تقسیم‌بندی می‌شوند. در روش ایستا تخصیص وظایف به ماشین‌های مجازی، براساس قابلیت‌های ماشین مجازی و وضعیت اولیه هر ماشین، می‌باشد، برخلاف الگوریتم‌های ایستا، در روش‌های پویا توزیع‌کننده علاوه بر قابلیت‌های اولیه هر ماشین مجازی، براساس وضعیت حال حاضر آن ماشین و بارکاری موجود بر آن، وظایف را به ماشین‌های مجازی اختصاص می‌دهد. روشی را که ما برای تعادل بار استفاده کرده‌ایم الهام گرفته از مقاله‌ای است که در سال ۲۰۱۳ در [۱۶] بر پایه رفتار زنبور عسل ارائه شده است، که هدف در آن کاهش زمان تکمیل تمامی کارها بود و در آن ابتدا درخواست‌ها به صورت نوبت گردشی به ماشین‌های مجازی اختصاص داده می‌شد، سپس درخواست‌ها از ماشین دارای اضافه بار به ماشین دارای بار کم انتقال داده می‌شد.

همانطور که گفتیم در رابطه با الگوریتم بهینه‌سازی ازدحام ذرات کارهای قبلی زیادی انجام شده است اما تفاوتی که در کار ما با سایر کارهای انجام شده وجود دارد این است که در روش پیشنهادی ما با ترکیب الگوریتم بهینه‌سازی ازدحام ذرات و الگوریتم تعادل بار، به منظور تولید نسل اولیه مناسب‌تر و انتخاب تابع هدف مناسب، از طرفی توانسته‌ایم با کاهش زمان تکمیل تمامی کارها و پاسخگویی سریع به درخواست‌های مشتری رضایت‌مندی مشتری را به همراه داشته باشیم و از طرف دیگر با افزایش بهره‌وری منابع،سبب افزایش رضایت‌مندی فراهم‌کننده‌ی سرویس شویم. نتایج شبیه‌سازی نشان می‌دهد که روش پیشنهادی در مقایسه با روش بهینه‌سازی ازدحام ذرات بدون استفاده از تعادل بار و روش بهینه‌سازی ازدحام ذراتی که نسل اولیه آن به کمک الگوریتم SJFP[۱۲] بهینه شده است، به صورت کاراتر عمل می‌کند.

۳- روش پیشنهادی

۳-۱- بهینه‌سازی ازدحام ذرات کلاسیک

الگوریتم بهینه‌سازی ازدحام ذرات یک الگوریتم بهینه‌سازی فرا اکتشافی است که از حرکت گروهی پرندگان (و دیگر حیواناتی که به شکل گروهی زندگی می‌کنند) الگو گرفته است. در این الگوریتم هر پاسخ مساله به صورت یک ذره که دارای یک مقدار و همچنین میزان تناسب است، مدل می‌شود. الگوریتم بهینه‌سازی ازدحام ذرات اولین بار توسط راسل و کندی در سال ۱۹۹۵ ارائه شد [۱۲]. الگوریتم بهینه‌سازی ازدحام ذرات بر مبنای حرکت و هوش ذرات کار می‌کند. در این الگوریتم هر ذره در حال جستجو برای نقطهٔ بهینه می‌باشد در نتیجه نیازمند آن است که در حال جابه‌جایی باشد، زیرا در غیر این‌صورت نمی‌تواند به فرایند جستجو ادامه دهد.

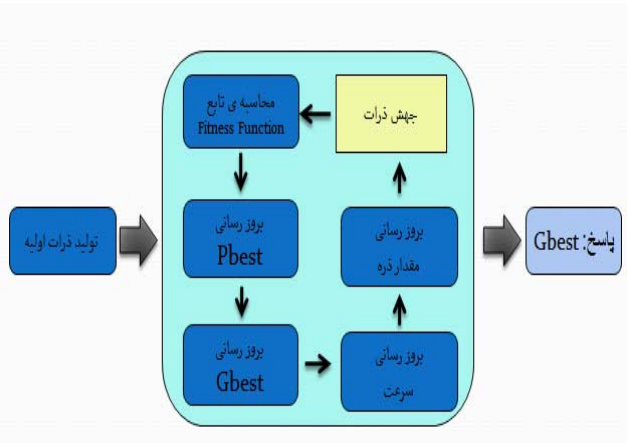
حرکت هر ذره به سه عامل بستگی دارد:

۱- موقعیت فعلی ذره.

۲- بهترین موقعیتی که تا کنون ذره داشته است. (بهترین موقعیت محلی[۱۳])

۳- بهترین موقعیتی که کل مجموعه‌ی ذرات تاکنون داشته‌اند. (بهترین موقعیت سراسری[۱۴])

در الگوریتم بهینه‌سازی ازدحام ذرات کارایی هر ذره براساس تابع برازندگی آن می‌باشد. هر ذره در هر مرحله، موقعیتی را که بهترین نتیجه را در آن داشته به خاطر می‌سپارد. (بهترین موقعیت فردی هر ذره) همچنین ذرات در گروه ذرات با هم همیاری می‌کنند. ذرات اطلاعاتی که دربارهٔ موقعیتی که در آن هستند را با هم تبادل می‌کنند. در نتیجه‌ی این تبادل اطلاعات بهترین موقعیت کل مجموع ذرات را می‌توان نگه‌داری کرد. در هر مرحله بر اساس اطلاعاتی که از بهترین موقعیتی که تاکنون ذره داشته است و بهترین موقعیتی که کل مجموعه‌ی ذرات تاکنون داشته‌اند می‌توانیم مقدار سرعت را به‌روزرسانی کنیم و سپس براساس سرعت به روز شده مقدار ذره را به روزرسانی کرده و سرانجام مقدار جهش ذره را تعیین کنیم. شکل ۱ مراحل الگوریتم بهینه‌سازی ازدحام ذرات را نشان می‌دهد.



شکل ۱- مراحل الگوریتم بهینه‌سازی ازدحام ذرات

در هر نسل سرعت و موقعیت ذره با استفاده از روابط (۱) و (۲) به روز می‌شود:

$$V_i^{k+1} = V_i^k + c_1.\mathrm{rand}_1 \times \left(\mathrm{pbest}_i - X_i^k\right) + c_2.\mathrm{rand}_2 \times \left(\mathrm{gbest}_i - X_i^k\right) \quad (۱)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (۲)$$

هر یک از متغیرهای روابط (۱) و (۲) مطابق جدول ۱ تعریف می‌گردد.

جدول ۱- تعریف متغیرهای الگوریتم PSO

v_i^k	سرعت ذره i در مرحله k
v_i^{k+1}	سرعت ذره i در مرحله k+1
w	وزن داخلی
c_j	ضریب شتاب j=1,2
$rand_i$	عدد تصادفی بین ۰و۱ i=1,2
x_i^k	موقعیت فعلی ذره i در مرحله k
$pbest_i$	بهترین موقعیت ذره i
$gbest_i$	بهترین موقعیت در بین تمامی ذرات
x_i^{k+1}	موقعیت فعلی ذره i در مرحله k+1

در ادامه روش پیشنهادی برای استفاده از الگوریتم بهینه‌سازی ازدحام ذرات به همراه الگوریتم تعادل بار، برای حل مسئله‌ی زمان‌بندی، بیان می‌شود.

۳-۲- استفاده از الگوریتم بهینه‌سازی ازدحام ذرات به‌همراه الگوریتم تعادل بار در مسئله‌ی زمان‌بندی

محاسبات ابری با تخصیص مجموعه‌ای از وظایف محاسباتی بر روی مجموعه‌ای از ماشین‌های مجازی بر طبق نیازمندی کاربران و ویژگی‌های هر یک از ماشین‌های مجازی سروکار دارد، درخواست‌های کاربران برای سرویس‌های مختلف بر طبق سیاست‌های زمان‌بندی مختلف به هر یک از ماشین‌های مجازی هدایت می‌شود. امروزه الگوریتم‌های مختلفی برای حل مسئله‌ی زمان بندی با اهداف مختلف ارائه شده است. هدف ما در این مقاله کاهش زمان تکمیل تمامی کارها و افزایش بهره‌وری منابع می‌باشد.

۳-۲-۱- مدل ریاضی الگوریتم

فرض کنید $VM = \{VM_1, VM_2, \dots, VM_m\}$ مجموعه‌ی ماشین‌های مجازی باشند که برای میزبانی درخواست‌های کاربران استفاده می‌شوند. همچنین $Task = \{T_1, T_2, \dots, T_n\}$ مجموعه‌ای از وظایفی می‌باشد که قصد اجرا بر روی ماشین‌های مجازی را دارد. هدف از الگوریتم کاهش در میزان زمان تکمیل تمامی کارها و افزایش متوسط بهره‌وری منابع می‌باشد که زمان تکمیل تمامی کارها را، طبق رابطه‌ی (۳) [۱۹] تعریف می‌کنیم:

$$\text{Makespan} = \max_{1\le i\le m} \sum_{j=1}^{n} PT_{ij} * x_{ij} \qquad (۳)$$

به‌طوریکه PT_{ij} در رابطه (۳) زمان تکمیل وظیفه j بر روی ماشین مجازی i می‌باشد. همچنین در صورتی‌که درخواست j بر روی ماشین مجازی i اجرا شده باشد مقدار xij برابر ۱ و در غیر این صورت برابر ۰ می‌باشد.

طبق رابطه (۳) از بین ماشین‌های مجازی ۱ تا m، ماشینی که مجموع زمان تکمیل تمامی وظایف محوله به آن، از همه بیشتر است، عامل تعیین makespn بوده و زمان تکمیل تمامی وظایف آن ماشین به عنوان makespan در نظر گرفته می‌شود.

درصورتی که مقدار بهره‌وری از هر منبع طبق رابطه‌ی (۴) [۱] محاسبه شود، متوسط بهره‌وری طبق رابطه‌ی (۵) محاسبه می‌شود.

$$\text{Utilization}_i = \frac{\sum_{j=1}^{n} PT_{ij} * xij}{\text{Makespan}} \qquad (۴)$$

به‌طوریکه PT_{ij} در رابطه (۴) زمان پردازش وظیفه j بر روی ماشین i می‌باشد.

در رابطه (۴) میزان بهره‌وری از منبع i برابر مجموع زمان پردازش تمام وظایف محول شده به ماشین مجازی i، به نسبت زمان پردازش ماشینی است که مجموع زمان پردازش تمامی وظایف آن از باقی ماشین‌ها بیشتر می‌باشد. بدیهی است که مقدار بهره‌وری از هر ماشین مجازی کمتر و یا مساوی عدد یک می‌باشد.

$$Average\ Utilization = \frac{\sum_{i=1}^{m} Utilization_i}{m} \qquad (۵)$$

در نتیجه تابع برازندگی با هدف کاهش زمان تکمیل تمامی کارها و افزایش بهره‌وری از منابع به صورت رابطه‌ی (۶) تعریف می‌شود:

$$Fitness\ Function = \frac{Makespan}{average\ Utilization} \qquad (۶)$$

با توجه به رابطه‌ی (۶) می‌توان نتیجه گرفت که هر چه قدر مقدار تابع برازندگی کمتر باشد میزان مطلوبیت ذره از جهت کاهش زمان تکمیل تمامی کارهاو افزایش بهره‌وری از منابع بیشتر است. در ادامه جزئیات الگوریتم پیشنهادی را به تفصیل بیان کرده و در انتها شبه کد آن را قرار می‌دهیم.

۳-۲-۲- جزئیات الگوریتم پیشنهادی تعادل بار با استفاده از الگوریتم ازدحام ذرات

- انتخاب ذرات اولیه و سرعت اولیه

ابتدا اندازه‌ی هر ذره را به تعداد وظایف قرار داده و برای هر ذره یک مکان تصادفی و یک سرعت اولیه تصادفی ایجاد می‌کنیم. برای مثال در صورتی که تعداد شش وظیفه و سه ماشین مجازی داشته باشیم ممکن است به صورت تصادفی وظایف طبق جدول ۲ بر روی ماشین‌های مجازی قرار بگیرند.

جدول ۲- نحوه قرارگیری مکان اولیه ذرات در الگوریتم PSO

وظیفه ۱	وظیفه ۲	وظیفه ۳	وظیفه ۴	وظیفه ۵	وظیفه ۶
ماشین مجازی ۲	ماشین مجازی ۱	ماشین مجازی ۱	ماشین مجازی ۳	ماشین مجازی ۲	ماشین مجازی ۲

- استفاده از روش تعادل بار برای متعادل کردن هر ماشین مجازی و بهبود مکان ذرات

در این مرحله میزان بار هر ماشین مجازی را طبق رابطه‌ی (۷) محاسبه کرده و ماشین‌های مجازی را به سه دسته‌ی ماشین های مجازی دارای افزونگی بار، ماشین‌های مجازی دارای کم‌باری و ماشین‌های مجازی دارای بارکاری متعادل تقسیم‌بندی می‌کنیم.

$$L_{V,M_i,t} = \frac{N(T,t)}{S(VM_i,t)} \qquad (۷)$$

بار هر ماشین مجازی براساس نسبت تعداد درخواست‌های موجود در صف سرویس‌دهی ماشین مجازی i در زمان t، برروی نرخ سرویس‌دهی ماشین i در زمان t محاسبه می‌شود.

برای این کار میزان بارکاری ماشین مجازی را در مقایسه با کل بار کاری موجود در تمامی ماشین‌های مجازی بررسی می‌کنیم، سپس براساس نتایج حاصل از مقایسه ماشینی که بارکاری آن از متوسط بارکاری سیستم بالاتر بود دارای افزونگی بار، در صورتی که در محدوده‌ی متوسط بارکاری سیستم بود دارای تعادل بار و در غیراین صورت دارای کم باری می‌باشد. سپس درخواست‌ها را از ماشین‌هایی که دارای افزونگی بار هستند یکی یکی برداشته و بر روی ماشین‌های کم بار قرار می‌دهیم. این کار را تا جایی ادامه می‌دهیم که یکی از مجموعه‌های ماشین‌های دارای افزونگی بار و یا ماشین‌های دارای کم باری خالی شود.

- محاسبه‌ی تابع برازندگی و انتخاب مقادیر Pbest, Gbest

در این مرحله مقدار تابع برازندگی را برای هر یک از ذرات طبق رابطه‌ی (۶) محاسبه می‌کنیم. مقدار بهترین موقعیت محلی را مقدار اولیه هر ذره قرار می‌دهیم و از بین ذرات، ذره‌ای که دارای بهترین مقدار تابع برازندگی می‌باشد را به عنوان بهترین موقعیت سراسری انتخاب می‌کنیم.

- به‌روزرسانی مکان ذرات و سرعت آنها

در این مرحله طبق رابطه‌ی (۱) ابتدا میزان سرعت اولیه براساس اطلاعاتی که از سرعت قبلی، بهترین مکان فعلی ذره و بهترین مکان در کل ذرات داریم، به‌روز می‌شود، سپس مکان ذره براساس سرعت به‌روز شده طبق رابطه‌ی (۲) تعیین می‌شود . سپس تابع برازندگی برای موقعیت جدید ذره محاسبه شده و در صورتی که مقدار آن از موقعیت قبلی ذره کمتر بود بهترین موقعیت محلی به ذره‌ی جدید تغییر می‌یابد. در بین گروه جدید مکان ذرات در صورتی کمترین مقدار تابع برازندگی از مقدار بهترین موقعیت سراسری قبلی کمتر بود، مقدار بهترین موقعیت سراسری به‌روزرسانی می‌شود.

- تکرار مراحل فوق به اندازه‌ی تعداد مراحل[15] الگوریتم

به تعداد مراحل الگوریتم، موقعیت هر ذره به روزرسانی شده و همچنین براساس موقعیت هر ذره و گروه ذرات مقادیر بهترین موقعیت محلی و بهترین موقعیت سراسری محاسبه می‌شود. در نهایت ذره‌ای که کمترین مقدار بهترین موقعیت سراسری را دارد به عنوان پاسخ مساله در نظر گرفته می‌شود. در ادامه شبه کد الگوریتم پیشنهادی در الگوریتم ۱ آورده شده است.

الگوریتم ۱- شبه‌کد الگوریتم پیشنهادی

Input: $VM = \{VM_1, VM_2, \dots, VM_m\}$,
$Task = \{T_1, T_2, \dots, T_n\}$
Output: best position of Tasks on the VM(Gbest)
Start:
1: Set particle dimensionas equal to the size of ready tasks
2: Initialize particles position randomly and velocity vi randomly.
3: for each particle run load balancing algorithm for balance particles position
4: For each particle, calculate its fitness value as in Equation(3-6)
  If (fitness value < previousbestpbest)
    set the current fitness value as the newpbest
5: After Steps 4 for all particles, select the best particle as gbest
6: For all particles, calculate velocity using Equation (3-1)
    and update their positions using Equation (3-2)
7: If (stopping criteria or maximum iteration is not satisfied)
  repeat from Step 4.
 else
  Return gbest
Stop.

با استفاده از الگوریتم ارائه شده می‌توان درخواست‌ها را به بهترین صورت ممکن به ماشین‌های مجازی اختصاص دهیم به‌طوری که میزان زمان تکمیل تمامی کارها، را کاهش داده و حداکثر استفاده از منابع را داشته باشیم. مزیت الگوریتم ارائه شده در روش فوق نسبت به روش پایه‌ی بهینه‌سازی ازدحام ذرات در این است، که در روش پایه‌ی بهینه‌سازی ازدحام ذرات موقعیت اولیه هر ذره به صورت تصادفی انتخاب می‌شود در صورتی که در روش ارائه شده موقعیت هر ذره در ابتدا به صورت تصادفی تعیین شده و سپس با استفاده از روش تعادل‌بار، برای توزیع عادلانه‌ی درخواست‌ها بین ماشین‌های مجازی مختلف، جابه‌جا می‌شود. با استفاده از این روش هر ذره‌ی اولیه خود یک ذره‌ی مناسب می‌باشد که با جابه‌جایی آن به جواب مناسب‌تر می‌رسیم. نتایج شبیه‌سازی نشان می‌دهد که روش فوق نسبت به روش پایه‌ی بهینه‌سازی ازدحام ذرات، بهبود قابل توجه‌ای را در زمان پاسخ و میزان بهره‌وری منابع دارد.

۴- شبیه‌سازی و ارزیابی روش پیشنهادی

در این بخش به جزئیات شبیه‌سازی الگوریتم معرفی شده در بخش قبل خواهیم پرداخت. سپس از طریق رسم نمودار به ارزیابی روش‌های مذکور می‌پردازیم. ذکر این نکته که محیط مورد بررسی ما یک محیط نرم‌افزار به‌عنوان سرویس است و همچنین ابزار ما برای شبیه‌سازی نرم‌افزار کلودسیم [۱۷] است، مفید خواهد بود. این شبیه‌ساز، به ما اجازه‌ی ایجاد یک محیط مجازی‌سازی شده را می‌دهد و از تخصیص منابع براساس تقاضا پشتیبانی می‌کند. در واقع ما هسته این شبیه‌ساز را برای مدل کردن الگوریتم ارائه شده گسترش داده‌ایم.

برای ارزیابی این بخش یک مرکز داده رایانش ابری را شبیه‌سازی کرده‌ایم که از سه میزبان تشکیل شده که قابلیت مجازی‌سازی[16] دارند. در واقع فرض شده است که روی آن‌ها مجازی سازهایی مثل Xen نصب شده است،که می‌تواند منابع را به اشتراک بگذارد. مشخصات هر یک از میزبان‌ها مطابق با جدول ۳ می‌باشد.

جدول ۳- مشخصات میزبان‌ها

شماره میزبان	تعداد هسته‌های پردازنده	سرعت پردازنده (تعداد دستورالعمل در ثانیه)	حافظه داخلی (مگا بیت)	حافظه جانبی (مگا بیت)	پهنای باند (مگا بیت بر ثانیه)
۱	۴	۵۰۰۰۰	۲۰۴۸۰۰	۱۰۴۸۵۷۶	۱۰۲۴۰۰
۲	۲	۲۵۰۰۰	۱۰۲۴۰۰	۱۰۴۸۵۷۶	۱۰۲۴۰۰
۳	۱	۱۰۰۰۰	۵۱۲۰۰	۱۰۴۸۵۷۶	۱۰۲۴۰۰

روی این مرکز داده ۱۶ ماشین مجازی با مشخصات متفاوت قرار داده ایم. هرکدام از ماشین‌های مجازی چند برنامه کاربردی با تعداد دستورالعمل‌های متغیر بین ۵۰۰ تا ۴۵۰۰ دستورالعمل را اجرا می‌کنند. همانطورکه قبلا هم اشاره کردیم هدف ارائه‌ی الگوریتم جامع و مناسب برای زمان‌بندی درخواست‌ها در بستر ابر می‌باشد به طوری‌که زمان تکمیل تمامی کارها[17] در آن کمترین بوده و حداکثر بهره‌وری از منابع را در بر داشته باشد. برای شبیه‌سازی الگوریتم بهینه‌سازی ذرات ده ذره را در نظر گرفته‌ایم که طی ۱۰۰ مرحله تکرار[18] به جواب مناسب می‌رسیم. پارامترهای لازم برای شبیه‌سازی الگوریتم ارائه شده مطابق با جدول ۴ می‌باشد.

در بخش اول مقایسه‌ای جهت بررسی زمان تکمیل تمامی کارها، بین روش‌های نوبت گردشی که بدون استفاده از روش‌های تعادل بار به توزیع درخواست‌ها می‌پردازد و سپس روش استفاده از الگوریتم بهینه‌سازی ازدحام ذرات پایه و روش تعادل بار بدون استفاده از بهینه‌سازی ذرات و الگوریتم بهینه‌سازی ازدحام ذرات بهبود یافته انجام می‌شود. نمودار شکل ۲ این مقایسه را

انجام می‌دهد. در این نمودار محور افقی تعداد وظایف و محور عمودی زمان تکمیل تمامی کارها، را نشان می‌دهد.

جدول ۴- پارامترهای لازم برای شبیه‌سازی

الگوریتم بهینه‌سازی ازدحام ذرات	اندازه جمعیت اولیه	۱۰
	تعداد تکرار	۱۰۰
	C_1,C_2	۱.۴۹۴۴۵
	r1 و r2	عدد تصادفی بین ۰ تا ۱
	W	w=0.99*w, Initial w=1
الگوریتم تعادل بار	محدوده متوسط بار کاری	meanload±0.05*meanload

برای شبیه‌سازی و مقایسه نتایج، از بارکاری حقیقی استفاده شده است این بار کاری مربوط به مرکز تحقیقات ناسا[19] می‌باشد که از اول ماه اکتبر[20] تا ۳۱ دسامبر[21] اندازه‌گیری شده است و شامل ۴۲۲۴۰ کار می باشد [۱۸].

هر کار موجود در این بار کاری شامل اطلاعات زیر می‌باشد:

- شناسه‌ی کار
- زمان ورود کار در سیستم
- زمان اجرای کار
- تعداد پردازنده‌های مورد نیاز برای یک کار
- تعداد کل پردازنده‌های مورد نیاز
- شناسه ی کاربری که کار را ایجاد کرده است
- شناسه‌ی گروه کاربرانی که کار را ایجاد کرده‌اند
- آرایه موقتی برای ذخیره‌ی فیلدهای هر کار

زمان اجرای هر کار به صورت متغیر از ۱ تا ۱۰۰۰۰ می‌باشد. برای استفاده از این فایل، آن را از ورودی گرفته و هر یک از کارها را به یک وظیفه در شبیه‌ساز کلودسیم به نام کلادلت تبدیل می‌کنیم. به این صورت که تعداد دستورالعمل‌های کلادلت را حاصلضرب زمان اجرای کار بر نرخ پردازنده[22] در نظر گرفته و یک کلادلت را می‌سازیم. برای ارزیابی از چهار ماشین فیزیکی با مشخصات جدول ۵ استفاده می‌شود:

جدول ۵- مشخصات ماشین فیزیکی مورد استفاده

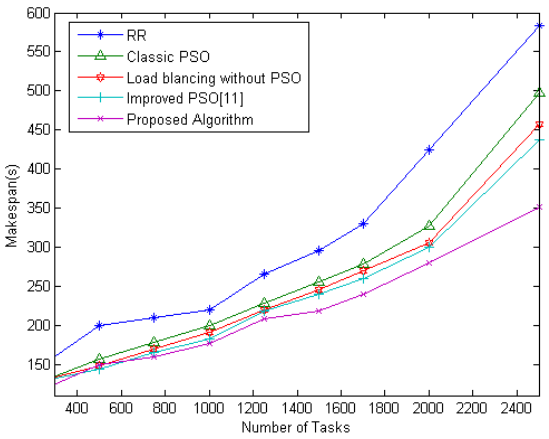
پردازنده	تعداد هسته‌های پردازنده	سرعت پردازنده (تعداد دستورالعمل در ثانیه)	حافظه داخلی (مگا بیت)	حافظه جانبی (مگا بیت)	پهنای باند (مگا بیت بر ثانیه)
Core_2_Extreme _X6800	۲	۲۷۰۷۹	۲۰۴۸۰	۱۰۴۸۵۷۶	۱۰۲۴۰۰
Core_i7_Extreme _Edition_X3960	۶	۱۷۷۷۳۰	۱۰۲۴	۱۰۴۸۵۷۶	۱۰۲۴۰۰
Core_i7_Extreme _Edition_980X	۶	۱۴۷۶۰۰	۲۰۴۸۰	۱۰۴۸۵۷۶	۱۰۲۴۰۰
Core_i7_875K	۴	۹۲۱۰۰	۲۰۴۸۰	۱۰۴۸۵۷۶	۱۰۲۴۰۰

همچنین به تعداد ۴۰ ماشین مجازی با مشخصات جدول ۶ برای ارزیابی روش ارائه شده استفاده می‌کنیم.

جدول ۶- مشخصات ماشین‌های مجازی مورد استفاده برای ارزیابی

پردازنده	تعداد هسته‌های پردازنده	سرعت پردازنده (تعداد دستورالعمل در ثانیه)	حافظه داخلی (مگا بیت)	حافظه جانبی (مگا بیت)	پهنای باند (مگا بیت بر ثانیه)
Core_i4_Extreme_Edition	۱	۹۷۲۶	۵۱۲	۱۰۲۴۰	۱۰۲۴

در نمودار شکل ۲ با ثابت نگه داشتن تعداد ماشین مجازی به میزان ۴۰ ماشین مجازی و افزایش تعداد درخواست‌ها از ۳۰۰ تا ۲۵۰۰ درخواست مقدار زمان تکمیل تمامی کارهارا در روش ارائه شده با سایر روش‌ها مقایسه کرده‌ایم.



شکل ۲- مقایسه makespan بین روش‌های مختلف

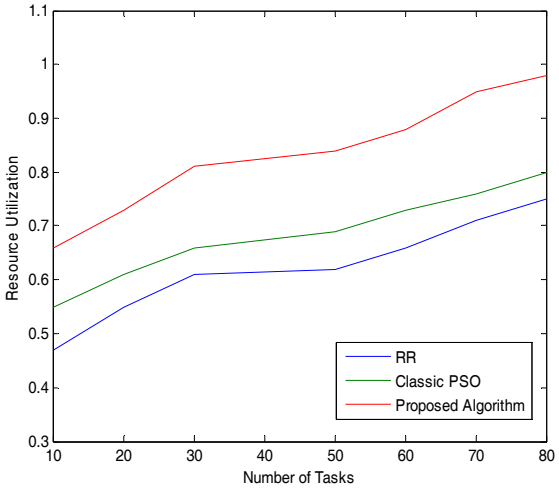
همانطور که در مجموعه‌ی نمودارهای شکل ۲ مشخص است استفاده از الگوریتم بهینه‌سازی ذرات باعث بهبود خوبی در زمان تکمیل تمامی کارها نسبت به روش پایه نوبت گردشی می‌شود. هم‌چنین استفاده از روش‌های تعادل بار نیز باعث کاهش زمان تکمیل تمامی کارها، نسبت به روش نوبت گردشی و روش پایه‌ی استفاده از الگوریتم بهینه‌سازی ذرات می‌شود. که این امر به دلیل تشخیص ماشین‌های مجازی دارای اضافه بار و کاهش بار آنها با جابه جایی بار بر روی ماشین‌های کم بار می باشد.

اما روش بهینه‌سازی ذرات که از ابتدا هر ذره در جایگاه مناسب قرار بگیرد از روش استفاده از الگوریتم تعادل بار مناسب تر است زیرا همانطور که در بخش قبل هم گفتیم کارایی الگوریتم تعادل بار وابسته به نحوه‌ی جایگزینی اولیه‌ی درخواست‌های آن مربوط می‌شود. در صورتی که درخواست‌ها را از ابتدا به‌صورت مناسب بر روی ماشین‌های مجازی قرار دهیم در هر لحظه جلوی افزونگی بار روی ماشین‌های مجازی گرفته می‌شود و بهبود قابل ملاحظه‌ای را نسبت به روش تعادل باری که جایگزینی اولیه درخواست‌ها به صورت نوبت گردشی باشد به دست می‌آوریم.

در روش پیشنهادی ابتدا در هر یک از ده ذره که یک جواب می‌باشد درخواست‌ها به‌صورت تصادفی بر روی ماشین‌ها قرار گرفته و سپس با استفاده از روش تعادل بار آنها را به‌منظور تعدیل بار و کاهش ماشین‌هایی که دارای افزونگی بار هستند، جابه‌جا می‌کنیم.

همانطور که در نمودار شکل ۲ مشخص است کارایی این روش با افزایش درخواست‌ها بهتر نیز می‌شود این امر به این دلیل می‌باشد که در درخواست‌های کم، بارکاری کم می‌باشد در نتیجه تعداد ماشین‌هایی که دارای افزونگی بار هستند، کمتر است، علاوه بر این مقدار زمان تکمیل تمامی کارها، بیشتر وابسته به تعداد درخواست‌ها و میزان زمان اجرای آن‌ها می‌باشد تا نحوه‌ی قراردهی درخواست‌ها بر روی ماشین‌های مجازی، در نتیجه با افزایش درخواست‌ها این بهبود به‌طور قابل ملاحظه افزایش می‌یابد.

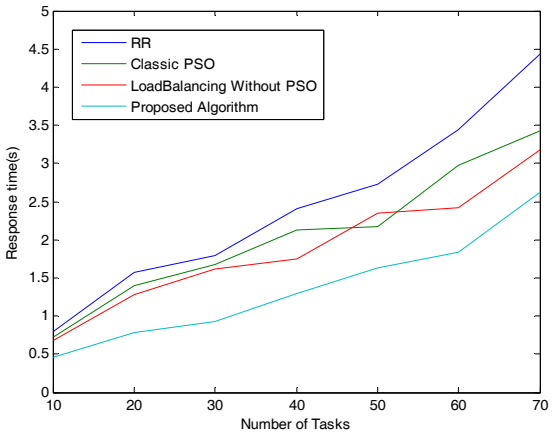
در ادامه به مقایسه‌ی بهره‌وری منابع بین روش‌های پایه، روش بهینه‌سازی ازدحام ذرات کلاسیک و روش بهینه‌سازی ازدحام ذرات بهبودیافته می‌پردازیم. شکل ۳ نمودار مقایسه‌ی بهره‌وری منابع را بین روش‌های بیان شده نشان می‌دهد. محور افقی در این نمودار تعداد درخواست‌ها و محور عمودی میزان بهره وری منابع می‌باشد.



شکل ۳- بهره‌وری منابع بین روش‌های PSO و PSO ترکیبی با روش تعادل بار

همانطور کـه از نمـودار شـکل ۳ مشـخص اسـت در هـر دو روش بـا افـزایش درخواست‌ها به دلیل تعادل بار ایجـاد شـده توسـط هـر یـک از روش‌هـای فـوق و افزایش بار کاری، بهره‌وری منابع به نسبت روش پایه بیشتر می‌شـود، امـا در روش بهینه‌سازی ذرات بهبود یافته نسبت به روش پایه چون میزان تعادل بار نسـبت بـه روش پایه بیشتر است، میزان بهبود در بهره‌وری منابع بیشتر می‌شود.

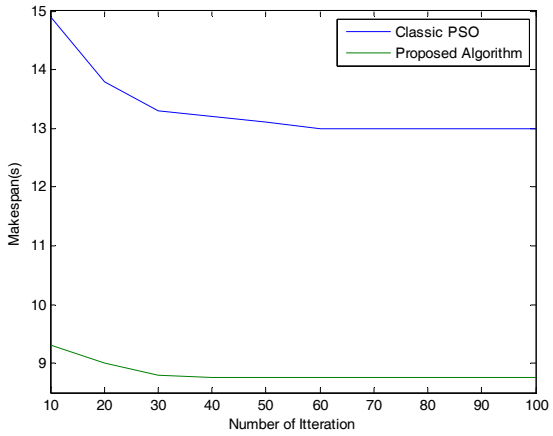
در ادامه متوسط زمان پاسخ را بین روش‌های مختلف مقایسه کرده‌ایم. شکل ۴ نمودار مقایسه‌ی متوسط زمان پاسخ را بین روش‌های بیان شـده نشـان مـی‌دهـد. محور افقی در این نمودار تعداد درخواست‌ها و محور عمودی میزان متوسـط زمـان پاسخ می‌باشد.



شکل ۴- مقایسه زمان پاسخ بین روش‌های مختلف

نمودار شکل ۵ میزان همگرایی تا رسیدن به جواب مناسب را با تعداد تکرار در هر مرحله نشان می‌دهد، تعداد درخواسـت‌هـا ۱۰۰ درخواسـت بـوده و مشخصـات شبیه‌سازی همانند شکل ۲ می‌باشد. همانطور که مشخص است، با افـزایش تعـداد مراحل تکرار، مقدار زمان تکمیل تمامی کارها، کاهش می‌یابد.

همان‌طور که در نمودارهای شکل ۵ مشخص است در روش بهینه‌سازی ازدحام ذرات چون جایگاه اولیه درخواست‌ها به صورت تصادفی انتخاب شده‌است با تعـداد تکرار ۳۰ تا ۴۰ مرحله می‌توانیم به جواب مناسب دست یابیم. امـا در روش بهبـود یافته‌ی بهینه‌سازی ازدحام ذرات چون جایگاه اولیه درخواست‌ها به صورت مناسـب انتخاب شده است، علاوه بر کاهش زمان تکمیل تمامی کارها با تعداد تکرار ۲۵ تـا ۳۰ مرحله و به طور سریعتر می‌توانیم به جواب مناسب برسیم.



شکل ۵- رابطه‌ی همگرایی و تعداد تکرارها

۵- نتیجه‌گیری و کارهای آینده

در این مقاله ما یک روش زمان‌بندی کارها بر پایه ترکیـب الگـوریتم بهینـه‌سـازی ازدحام ذراتبه همراه الگوریتم تعادل بار ارائه کرده‌ایـم. ایـن الگـوریتم نـه تنهـا بـا متعادل کردن درخواست‌های ماشین‌های مجازی سبب کاهش زمان تکمیل تمامی کارها شده است، همچنین با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات توانسـته با انتخاب تابع برازندگی مناسـب، سـبب افـزایش بهـره‌وری منـابع شـود. مـا روش پیشنهادی خود را با روش پایه نوبت گردشی و هریک از الگوریتم‌های تعادل بـار و الگوریتم بهینه‌سازی ازدحام ذرات به تنهایی مقایسه کرده‌ایـم نتـایج شـبیه‌سـازی نشان می‌دهد که روش پیشنهادی بهبود خوبی را در زمان تکمیل تمـامی کارهـا و افزایش بهره‌وری منابع می‌دهد. در آینده قصد داریم تـا ایـن کـار را بـرای جریـان گردش کار نیز انجام دهیم همچنین معیارهای دیگر مثـل قابلیـت تحمـل خرابـی برای کاربر و کاهش هزینه را برای فراهم کننده‌ی سرویس در نظر بگیریم.

مراجع

[1] K. Krishnasamy, "Task Scheduling Algorithm Based on Hybrid Particle Swarm Optimization in Cloud Computing Environment," Journal of theoretical & applied information technology, vol. 54, 2013.

[2] P. Kumar, and A. Verma, "Scheduling Using Improved Genetic Algorithm In Cloud Computing for Independent Tasks," Proc. International Conference on Advances in Computing, Communications and Informatics, pp. 137-142, 2012.

[3] C. Zhao, S. Zhang, Q. Liu, J. Xie, and J. Hu, "Independent Tasks Scheduling Based on Genetic Algorithm in Cloud Computing," 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom'09), pp. 1-4, 2009.

[4] L. Wang, and L. Ai, "Task Scheduling Policy Based on Ant Colony Optimization in Cloud Computing

[16] P. V. Krishna, "Honey Bee Behavior Inspired Load Balancing of Tasks in Cloud Computing Environments," Applied Soft Computing, vol. 13, pp. 2292-2303, 2013.

[17] R. N. Calheiros, and et. al., "Cloudsim: A Toolkit For Modeling And Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," Software: Practice and Experience, vol. 41, pp. 23-50, 2011.

[18] D. G. Feitelson, and B. Nitzberg, "Job Characteristics of a Production Parallel Scientific Workload on the NASA Ames Ipsc/860," in Job Scheduling Strategies for Parallel Processing. Berlin, Germany: Springer Berlin Heidelberg, 1995, pp. 337-360.

[19] K. Gomathi, and B. Krishnasamy, "Task Scheduling Algorithm Base on Hybrid Particle Swarm Optimization in Cloud Computing Environment," Journal of Theoretical and Applied Information Technology, vol. 55, 2013.

فاطمه عبادی‌فرد مدرک کارشناسی مهندسی کامپیوتر گرایش نرم‌افزار را از دانشگاه قم اخذ کرده و در رشته مهندسی فناوری اطلاعات گرایش شبکه‌های کامپیوتری در مقطع کارشناسی‌ارشد، از دانشگاه علم و صنعت تهران فارغ‌التحصیل شده است و در حال حاضر دانشجوی دکتری مهندسی کامپیوتر در دانشگاه کاشان می‌باشد.

آدرس پست‌الکترونیکی ایشان عبارت است از:

f_ebadifardr@gmail.com

احمد اکبری دارای دکتری برق و کامپیوتر از دانشگاه رن فرانسه بوده و در حال حاضر دانشیار و عضو هیئت علمی دانشکده مهندسی کامپیوتر دانشگاه علم و صنعت ایران می‌باشد. ایشان پژوهش‌های فعالی را در زمینه‌های شبکه‌های کامپیوتری و امنیت و نیز پردازش سیگنال هدایت می‌کنند. همچنین مسئولیت ریاست دانشکده مهندسی کامپیوتر این دانشگاه و مرکز تحقیقات فناوری اطلاعات دانشگاه را نیز برعهده دارند.

آدرس پست‌الکترونیکی ایشان عبارت است از:

akbari@iust.ac.ir

اطلاعات بررسی مقاله:

تاریخ ارسال: ۱۳۹۴/۰۷/۲۰

تاریخ اصلاح: ۱۳۹۴/۰۹/۱۸

تاریخ قبول شدن: ۱۳۹۴/۱۰/۱۵

نویسنده مرتبط: فاطمه عبادی‌فرد، دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران.

[5] S. K. Chaharsooghi, and A. H. M. Kermani, "An Effective Ant Colony Optimization Algorithm (ACO) for Multi-Objective Resource Allocation Problem (MORAP)," Applied Mathematics and Computation, vol. 200, pp. 167-177, 2008.

[6] E. Pacini, C. Mateos, and C. G. Garino, "Balancing Throughput and Response Time in Online Scientific Clouds via Ant Colony Optimization," Advances in Engineering Software, vol. 84, pp. 31-47, 2015.

[7] L. Guo, S. Zhao, S. Shen, and C. Jiang, "Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm," Journal of Networks, vol. 7, pp. 547-553, 2012.

[8] L. Zhang, and et. al., "A Task Scheduling Algorithm Based on PSO for Grid Computing," International Journal of Computational Intelligence Research, vol. 4, pp. 37-43, 2008.

[9] S. Pandey, and et. al., "A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments," Proc. 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 400-407, 2010.

[10] S. Wang, and B. Meng, "Resource Allocation and Scheduling Problem Based on Genetic Algorithm and Ant Colony Optimization," in Advances in Knowledge Discovery and Data Mining. Berlin, Germany: Springer Berlin Heidelberg. 2007, pp. 879-886.

[11] S. Zhan, and H. Huo, "Improved PSO-based Task Scheduling Algorithm in Cloud Computing," Journal of Information & Computational Science, vol. 9, pp. 3821-3829, 2012.

[12] R. C. Eberhart, and J. Kennedy, "A New Optimizer Using Particle Swarm Theory," Proc. sixth International Symposium on Micro Machine and Human Science, pp. 39-43, 1995.

[13] G. Kaur, and E. S. Sharma, "Optimized Utilization of Resources Using Improved Particle Swarm Optimization Based Task Scheduling Algorithms in Cloud Computing," International Journal of Emerging Technology and Advanced Engineering, vol. 4, 2014.

[14] S. A. M. S. Abdi, and S. Sharifian, "Task Scheduling Using Modified PSO Algorithm in Cloud Computing Environment," Proc. International Conference on Machine Learning, Electrical and Mechanical Engineering, Dubai (UAE), 2014.

[15] K. A. Nuaimi, and et. al., "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms," Proc. Second Symposium on Network Cloud Computing and Applications (NCCA), pp. 137-142, 2012.

[1] Paticle Swarm Optimization
[2] Makespan
[3] Infrastructure As A Service (Iaas)
[4] Platform As A Service (Paas)
[5] Software As A Service (SaaS)
[6] Multimedia Application Provider

[7] Resource Utilizatio
[8] Load Balancing
[9] Best Resource Selection
[10] Static
[11] Dynamic
[12] Shortest Job to Fastest Processor Algorithm
[13] Pbest
[14] Gbest
[15] Iteration
[16] Virtualization
[17] Makespan
[18] Iteration
[19] NASA Ames Research Center
[20] October
[21] December
[22] CPU Rating