ارائه یکٔ الگوریتم زمان‌بندي وظایف کارا در محیطهای رایانش ابري بر مبنای الگوریتم ژنتیک بهینه سازی شده

سجاد میر [1] ، مهدی فاضلی [2]

[1] دانشجوی کارشناسی ارشد گروه کامپیوتر، واحد بروجرد، دانشگاه آزاد اسلامی، بروجرد، ایران، Sajjad.mir85@yahoo.com

[2] دکتری گروه کامپیوتر، واحد بروجرد، دانشگاه آزاد اسلامی، بروجرد، ایران، m-fazeli@iust.ac.ir

چکیده

رایانش ابری، مدلی کامپیوتری است که دسترسی کاربران را بر اساس نوع تقاضایی که از منابع اطلاعاتی و محاسباتی دارند محیا می‌کند . این مدل سعی دارد با کمترین نیاز به منابع و کاهش هزینه‌ها و افزایش سرعت دسترسی اطلاعات، جوابگوی نیاز کاربران باشد . مدیریت منابع و تخصیص وظایف در کنار امنیت، قابلیت اطمینان و حفظ اعتماد مشترکین یکي از چالش‌هاي مهم در زمینه رایانش ابري مي‌باشد که مي‌تواند روي سایر مسائل نیز تاثیرگذار باشد.واضح است که تعداد کارها و تعداد منابع در محیط ابر مي‌تواند بسیار گسترده باشد و به همین علت ترتیب اجراي کارها و نحوه تخصیص منابع تاثیر مهمي برکارآیي سرور ابر دارد. در محیط ابر درهر لحظه بصورت همزمان تعدادي درخواست داریم که براي پاسخ به هریکٔ از آنها باید یکٔ یا چند کار انجام دهیم و هریکٔ از این کارها به منابع خاصي نیاز دارند، که در این رابطه روش‌های مختلفی ارائه شده است. در این مقاله یک الگوریتم زمان‌بندی وظایف برای محیطهای ابری بر اساس الگوریتم ژنتیک بهینه‌سازی شده ارائه شده است که می‌تواند توازن بار را در سرور برقرار کرده و هزینه را بهینه کند. نتایج شبیه‌سازی نشان می‌دهد که روش پیشنهادی کارآیی بهتری نسبت به سایر الگوریتم‌ها دارد.

کلمات کلیدی

رایانش ابری، مدیریت منابع، الگوریتم ژنتیک، زمان پاسخ

۱- مقدمه

دنیای فناوری اطلاعات و اینترنت که امروزه تبدیل به جزئی از زندگی بشر شده روز به روز در حال گسترش است. همسو با آن، نیازهای اعضای جوامع مانند امنیت اطلاعات، پردازش سریع، دسترسی پویا و آنی، قدرت تمرکز روی پروژه‌های سازمانی به جای اتلاف وقت برای نگهداری سرورها و از همه مهمتر، صرفه‌جویی در هزینه‌ها اهمیت زیادی یافته است . راه‌حلی که امروزه در عرصه‌ی فناوری برای چنین مشکلاتی پیشنهاد می‌شود تکنولوژی است که در این روزها با نام رایانش ابری به آن پرداخته می‌شود. در رایانش ابری کاربران نیازی به سیستم‌های سخت‌افزاری با قابلیت پردازش و ظرفیت ذخیره‌سازی بالا ندارند، زیرا تمامی اعمال محاسباتی و ذخیره‌سازی در بخش ارائه‌دهندگان خدمات ابری و توسط سرورهای مجهز و پیشرفته انجام می‌گیرد. مسئله تخصیص منبع و زمانبندی در رایانش ابری یکی از چالش‌های زمینه رایانش ابری است. در این رابطه روش‌های مختلفی ارائه شده است. با این حال استفاده از تجارب قبلی در راستای بهبود الگوریتم زمانبندی مورد توجه است. این موضوع می‌تواند به بهبود مداوم الگوریتم منجر شود.

با توجه به وجود بسترهای رایانش ابری، جهت استفاده از امکانات رایانش ابری، مسئله زمانبندی اهمیت پیدا می کند و در نتیجه ارائه یک الگوریتم زمانبندی کارا در محیطهای رایانش ابری ضرورت پیدا می کند.

یکی از مهمترین قسمت‌های یک سرور رایانش ابری که می توان آن را بهبود داد و بهبود آن تأثیر بسیاری بر افزایش راندمان و افزایش کیفیت سرویس ها دارد، الگوریتم و روش تخصیص منابع به درخواست‌ها می‌باشد. ساختار و معماری سرور رایانش ابری به این صورت است که هر سرور به چند Host تقسیم شده و هر Host نیز از چند Virtul Machin تشکیل شده است. نحوه تخصیص منابع به این صورت است که پس از دسته بندی و آماده کردن درخواست‌ها برمبنای یک الگوریتم مشخص کارها (task) را به Virtul Machine(VM) مشخص شده توسط الگوریتم می‌فرستند و درواقع منابعی که در آن VM قرار دارد را به آن کار اختصاص می دهند. همچنین این نکته را نیز باید درنظر گرفت که هر VM باتوجه به توان و منابع خود می تواند چند کار را به صورت همزمان میزبانی کرده و انجام‌دهد.

وقتی کار بر یک سرویس خاص را از سرور ابر درخواست می‌کند مراحل ارائه آن سرویس توسط سرور بررسی‌شده و به چندین کار (task) وابسته یا غیروابسته تجزیه می شود و برای انجام آماده‌می‌شوند. در این تحقیق یک الگوریتم برای مدیریت منابع و تخصیص آنها به کارها (درخواست‌ها) یا به عبارتی یک الگوریتم برای تخصیص VM ها به task ها ارائه شده که در واقع یک مدل بهینه از الگوریتم ژنتیک می‌باشند. در واقع هدف نهایی این راه-دحل پیشنهادی به دست آوردن یک الگوی مناسب جهت نگاشت کار (task)

Archive of SID

www.SID.ir

به ماشین مجازی (VM) می‌باشد که بتواند بدون افزایش توان مصرفی زمان پاسخ و زمان تکمیل آخرین کار را به حداقل برساند.

۲- الگوریتم‌های تخصیص منبع در رایانش ابری

از آنجائی که تقاضای منبع و تخصیص آن می‌تواند پویا باشد، الگوریتم‌های مختلفی جهت تخصیص منبع پیشنهاد شده است. قرتر دادن الگوریتم‌های مناسب جهت گارانتی هوشمندانه و استفاده بهینه از منابع، مسئله‌ای مهم است. تخصیص منبع و زمانبندی، کلید انجام پردازش در رایانش ابری است.
به طور کلی می‌توان الگوریتم‌های تخصیص منبع را به دو دسته تقسیم کرد:

۱. الگوریتم‌های حریصانه
۲. الگوریتم‌های تکاملی

در ادمه به بررسی هر کدام از این دسته‌ها پرداخته و الگوریتم‌های تخصیص منبع موجود در هر دسته را تشریح می‌کنیم.

۲-۱- الگوریتم‌های حریصانه

روش حریصانه یکی از روش‌های مشهور و پرکاربرد طراحی الگوریتم است که با ساختاری ساده در حل بسیاری از مسائل استفاده می‌شود. این روش اغلب در حل مسائل بهینه‌سازی استفاده شده و در پاره‌ای مواقع جایگزین مناسبی برای روش‌هایی مانند برنامه‌ریزی پویا است. در حالت کلی این روش سرعت و مرتبه اجرائی بهتری نسبت به روش‌های مشابه خود دارد اما متناسب با مسأله ممکن است به یک جواب بهینه سراسری ختم نشود.
از الگوریتم‌های حریصانه می‌توان به الگوریتم‌های FCFS، Round Robin، چرخشی، Least-load و Ranking اشاره کرد که در اینجا استفاده از الگوریتم Round Robin جهت تخصیص منابع را با توجه به کاربرد آن در بعضی از سرورهای ابر و همچنین استفاده آن در شبیه‌سازی جهت مقایسه توضیح می‌دهیم.

۲-۱-۱- الگوریتم Round Robin

یک روش تصادفی برای انتخاب منبع فیزیکی توافقی برای تقاضای ماشین مجازی است. در این الگوریتم بدون توجه به اولویت کارها، به صورت چرخشی یک بازه زمانی ثابت را به تمام کارها اختصاص می‌دهد. از این روش در سیستم‌هایی چون Eucalyptuses استفاده شده است .
در [۹] مشخص شد که الگوریتم Round Robin با روش FCFS زمان پاسخ طولانی را بدست می‌آورد و همچنین توازن بار در منابع در نظر گرفته نمی‌شود. در این روش حداکثر استفاده از منبع فیزیکی انجام نمی‌شود. همچنین این روش نمی‌تواند به QOS مناسبی دست یابد.

۲-۲- الگوریتم‌های تکاملی

الگوریتم‌های تکامل‌پذیر روش‌هایی بر مبنای جستجوی تصادفی هستند که از مدل‌سازی تکامل بیولوژیکی طبیعی الگوبرداری شده‌اند. آن‌ها برروی پاسخ-های ممکنی کار می‌کنند که از ویژگی‌های برتری برخوردارند و بقای نسل بیشتری دارند، لذا تخمین نزدیک‌تری به پاسخ بهینه را بدست می‌آورند. در این الگوریتم‌ها در هر نسل، دسته جدیدی از تخمین‌ها بر مبنای انتخاب اعضاء با میزان شایستگی بیشتر تولید شده و سپس مشابه آنچه در طبیعت رخ می-دهد با هم تلفیق می‌شوند. در نتیجه این روند باعث تکامل اعضاء می‌شود و هر عضو نسبت به والدینشان در محیط سازگارتر می‌شود.
الگوریتم‌های تکامل‌پذیر در گستره وسیعی از مسائل کارآمد می‌باشند.از الگوریتم‌های تکاملی می‌توان به الگوریتم‌های ژنتیک، کلونی زنبور عسل، PSO، سرد و گرم کردن فلزات و رقابت استعماری اشاره کرد که در ادامه الگوریتم ژنتیک به عنوان یکی از الگوریتم‌های تکاملی پرکاربرد توضیح داده خواهدشد.

۲-۲-۱- الگوریتم ژنتیک

در این الگوریتم ابتدا هزینه ژن از طریق نسبت راه‌حل زمان‌بندی فعلی به بهترین راه‌حل زمان‌بندی محاسبه می‌شود سپس برای استراتژی زمانبندی مطابق با هزینه ژن تصمیم گرفته می‌شود. در نهایت راه‌حل زمان‌بندی که با کمترین هزینه، مشابه راه‌حل زمان‌بندی نهائی باشد انتخاب می‌شود. جهت تست کردن توازن بار الگوریتم ژنتیک، نگاشت وابستگی قبل و بعد از استفاده از الگوریتم را به ترتیب می‌گیریم. نتایج نشان می‌دهد که بعد از استفاده از الگوریتم ژنتیک، بارهای هر گره به متوازن شدن گرایش دارند و اختلاف بار سیستم بسیار کوچک است. بنابراین می‌توان نتیجه گرفت که الگوریتم توازن بار خوبی دارد و می‌تواند در یک زمان خیلی کوتاه به بهترین راه‌حل همگرا شود. وقتی که بارگذاری سیستم نسبتاً ثابت است (تغییرات بارگذاری سیستم نسبتاً کوچک است) الگوریتم ژنتیک به خوبی قادر است توازن بار سیستم را تضمین کند و زمانی که تغییرات بارگذاری سیستم نسبتاً بزرگ است، الگوریتم ژنتیک می‌تواند توازن بارسیستم را بهتر از روش‌های دیگر تضمین کند. نرخ نوسان بارگذاری ماشین‌های مجازی تأثیر مهمی در توازن بار سیستم دارد.
پس از زمان‌بندی منابع، تغییرات بزرگی در بارگذاری هر ماشین فیزیکی به وجود خواهد آمد و ممکن است سیستم در یک دوره زمانی به یک مهاجرت پویا جهت تحقق یافتن توازن بار بعد از زمانبندی نیاز داشته باشد. باید توجه داشت که از هزینه مهاجرت ماشین مجازی نمی‌توان غفلت کرد. بنابراین جائی که ماشین مجازی باید مهاجرت کند، چگونگی مهاجرت کمترین تعداد ماشین مجازی از مسائلی است که باید در طی زمانبندی ماشین مجازی ملاحظه شود.
آزمایشات انجام شده در [۸] روی الگوریتم ژنتیک نشان می‌دهند که تغییرات بارگذاری تأثیر کوچکی روی نتیجه توازن بار الگوریتم ژنتیک دارد. همچنین نمایش می‌دهد، وقتی که تغییرات بارگذاری سیستم نسبتاً کوچک است، هزینه مهاجرت رضایت‌بخش است و هزینه مهاجرت ایجاد شده به وسیله تغییرات سیستم بعد از زمانبندی کم است. همچنین نتایج نشان می‌دهند که در الگوریتم ژنتیک نسبت مهاجرت ماشین مجازی وقتی که نرخ تغییرات بارگذاری ماشین مجازی در حال تغییر است بسیار پائین است، این مسئله موجب کاهش هزینه مهاجرت می‌شود.
در الگوریتم ژنتیک با توجه به اطلاعات گذشته و وضعیت فعلی تأثیری که زمانبندی خواهد داشت محاسبه می‌شود، سپس حالتی که کمترین تأثیر را بر روی سیستم خواهد گذاشت به عنوان زمانبندی نهائی انتخاب می‌شود. بنابراین این روش کمترین تأثیر را بر روی بار سیستم بعد از زمانبندی دارد و همچنین کمترین هزینه جهت رسیدن به توازن بار را دارد. همچنین در

Archive of SID

www.SID.ir

الگوریتم ژنتیک تا زمانی که منابع کافی وجود دارد همیشه یک زمانبندی خوب ارائه می‌شود در نتیجه بهره‌وری این الگوریتم مناسب می‌باشد. پس در کل می‌توان گفت که الگوریتم ژنتیک روشی مناسب است.

۳- الگوریتم پیشنهادی

زمان‌بندی و مدیریت منابع در محیط های توزیع شده مانند سرورهای رایانش ابری به پارامترهای زیادی بستگی دارد، زیرا در این سیستم ها، منابع و درخواست های متفاوت، پویا و گاهاً وابسته بهم وجود دارند و همچنین حجم کاری سیستم نسبت به سایر محیط ها بیشتر می باشد. با توجه به اینکه مهمترین پارامترها در سرورهای رایانش ابری پارامترهایی هستند که بر زمان تکمیل درخواست‌ها و توزیع متناسب بار مبتنی می‌باشند، ما در این تحقیق روی روشی کار می‌کنیم که بتواند پارامترهای زمان پاسخ (Response time) و زمان تکمیل آخرین کار (Make span) که در واقع نشان دهنده توزیع متناسب بار می‌باشد را نسبت به الگوریتم‌های ارائه شده قبلی کاهش دهد.

برای انجام این کار ما از یک الگوریتم ترکیبی مبتنی بر الگوریتم ژنتیک استفاده می کنیم که به نحوی می‌توان گفت یک مدل بهینه از الگوریتم ژنتیک می‌باشد. دلیل انتخاب الگوریتم ژنتیک از بین الگوریتم‌های تکاملی، امکان اجرای موازی، سادگی محاسبه و دقت بالای آن می‌باشد.

برای بررسی کامل این الگوریتم یک سری پیش‌فرض در نظر گرفته‌شده که در ادامه به آنها اشاره شده است.

شرایط کارها (taskها) در جدول (۱) مشخص شده اند :

Dead line	نوع task	زمان مورد نیاز برای تکمیل	منبع مورد نیاز
ندارند	مستقل	۵۰۰ ~ ۳۵۰۰	پردازنده

جدول (۱) مشخصات taskها

همانطورکه در جدول مشخص شده است، نوع task ها پردازش بوده و به هم وابسته نیستند، همچنین شرایط ماشین های مجازی نیز در جدول (۲) مشخص شده است :

قابلیت میزبانی بیش از یک task	تعداد VM	ظرفیت VM	نوع منبع VM
✔	۱۰۰۰~۲۰۰	۱۰۰۰ ~ ۷۰۰۰	پردازنده

جدول (۲) مشخصات VMها

پس از انجام عملیات‌های اولیه روی taskها و دسته‌بندی و آماده‌سازی آنها، الگوریتم به صورت موازی روی هر دسته و ماشین‌های مجازی مربوط به آن اجرا می شود. به صورت خلاصه الگوریتم موردنظر به صورت زیر می‌باشد :

۳-۱- نحوه نمایش راه حل ها :

هر راه حل یا مدل تخصیص منابع به صورت یک رشته $n \times ۲$ نمایش داده می شود و هر رشته به عنوان یک کروموزوم الگوریتم ژنتیک در نظر گرفته می‌شود. شکل زیر یک راه حل را نشان می‌دهد.

T12	T23	T80	T8	T7	T2	T41	…	T6	T18
R2	R4	R6	R8	R10	R6	R3	…	R8	R4

جدول (۳) یک راه حل نمونه

در هر مدل تمامی task ها به VM انتخاب شده نگاشت می شوند، طول کروموزوم به اندازه n می باشد (که n برابر تعداد taskها است) و هر ستون از آن در واقع یک ژن می باشد.

۳-۲- ایجاد جمعیت اولیه

جمعیت اولیه متشکل از تعدادی راه حل می باشد که در الگوریتم ژنتیک استاندارد، به صورت تصادفی ایجاد می‌شود. در این راه‌حل ما قصد داریم جمعیت اولیه را به صورت هدفمند ایجاد نماییم تا علاوه بر تسریع در یافتن جواب، پاسخ به دست آمده نیز نسبت به حالت معمول، به پاسخ بهینه نزدیک‌تر باشد. روش کار به اینصورت است که برای نگاشت یک task به VM شرایط این عملیات طبق یک فرمول سنجیده می‌شود و در صورت انتخاب این نگاشت به عنوان یک ژن از کروموزوم راه‌حل اولیه مشخص مس‌شود و به همین ترتیب برای سایر taskها این عملیات انجام می‌شود تا یک کروموزوم به عنوان یکی از راه‌حل های ایجاد شود.

همان طورکه بیان گردید هر ژن نشان دهنده یک task و منبع مناسب برای آن می باشد. بهترین منبع از لحاظ زمان اجرای task برای آن در نظر گرفته می شود. در صورتی که منبعی واجد شرایط نباشد منبع بعدی (دومین بهترین منبع) تست می شود و این روند برای تمامی ژن ها صورت می‌گیرد و سپس لیست مجازی منابع کروموزوم براساس باقی مانده ظرفیت هر منبع با توجه حجم کاری taskها، به روز می شود. به دلیل کمتر بودن تعداد منابع نسبت به درخواست ها، هر منبع می بایست بیش از یک درخواست را پردازش نماید.

در این روش مناسب بودن یک VM جهت اختصاص به task مورد نظر طبق فرمول (۱) و در صورت برقرار بودن شرط، طبق فرمول (۲) بررسی می-شود:

$$\text{cloudletLength} > \text{freemips}، \quad (1)$$

$$rt = \text{alocatedmips} \quad (2)$$

که در آن:

rt : عدد معیار جهت انتخاب یا عدم انتخاب VM

Archive of SID

www.SID.ir

cloudletLength : طول task

freemips : مقدار منابع آزاد VM

alocatedmips : مقدار منابع اختصاص داده شده VM (فضای اشغال شده)

در این روش و روش‌های بعد هرچقدر مقدار rt کوچکتر باشد، شانس VM جهت انتخاب بیشتر می‌شود.

۳-۳- تابع برازندگی (fitness)

برای تشخیص اینکه یک راه حل، تا چه اندازه می‌تواند پاسخی مناسب ارائه دهد، می‌بایست ارزش آن توسط یک تابع fitness سنجیده شود، این تابع براساس پارامترهای مؤثر در کیفیت راه حل، یک ارزش را به راه حل نسبت می‌دهد. در الگوریتم ژنتیک با اعمال این تابع بر روی تمامی راه حل ها، ارزش هر کدام محاسبه گردیده و راه حل دارای بهترین ارزش که می‌تواند براساس سیاست قرارگیری پارامترها مقدار بیشینه یا کمینه داشته باشد، به عنوان مناسب ترین راه حل در نظر گرفته می‌شود. در این روش به منظور کاهش زمان makespan این فاکتور به عنوان fitness factor در نظر گرفته شده است.

به همین منظور این فاکتور را برای تمام کروموزوم‌ها محاسبه کرده و کروموزوم دارای کم ترین مقدار fitness، به عنوان راه حل مناسب تر در بین سایرین انتخاب می گردد. بنابراین :

Target is : Minimum ($Fitness_{total}$)

در طی اجرای الگوریتم با انجام مکرر عملیات crossover و mutation سعی در یافتن راه حل با مقدار fitness کمتر می شود.

۳-۴- عملیات ترکیب Crossover

دلیل اینکه هر منبع می تواند بیش از یک task را اجرا نماید لذا ترتیب قرار گیری و نگاشت task ها در منابع می‌تواند در میزان توازن بار و در نتیجه دریافت پاسخ مجموعه task ها در مدت زمان کوتاه‌تر مؤثر باشد و توسط عملیات crossover می توان شانس جابجایی ژن های را برای دستیابی به توازن بار بیشتر در راه حل افزایش داد.

در الگوریتم ژنتیک استاندارد برای انجام عملیات crossover دو کروموزوم به صورت تصادفی انتخاب می شود و یک ژن از آن ها به صورت تصادفی در نظر گرفته می شود [۱۴]. با جابجا کردن ژن های دو کروموزوم، fitness آنها محاسبه می گردد. در صورتی که مقدار fitness کروموزوم های جدید نسبت به والدین بیشتر باشد، آنها جایگزین می شوند. در غیر این صورت دو ژن دیگر انتخاب می گردند. این روند تا تعداد تکرار معینی ادامه می یابد. تفاوت crossover این روش‌ها با ژنتیک استاندارد در اینجاست که در این روش‌ها ابتدا میانگین fitness تمام کروموزوم‌های کاندید محاسبه شده سپس از بین آنهایی که fitness بیشتری از fitness میانگین دارند به صورت تصادفی دو کروموزوم جهت انجام عملیات crossover انتخاب می‌شود.

۳-۵- عملیات جهش (Mutation)

یک راه‌حل و یک ژن از آن راه‌حل به صورت تصادفی انتخاب می‌شوند. سپس از میان منابع موجود یک منبع را به صورت تصادفی در نظر می‌گیریم. مناسب بودن منبع براساس شرایط ذکر شده سنجیده می‌شود. در صورت برقرار نبودن شرایط، منبع دیگری انتخاب می گردد. این عمل سبب می شود تا الگوریتم در نقطه کمینه محلی به دام نیفتد.

با هر تغییر ژن در کروموزوم در عملیات crossover و mutation، لیست مجازی توان پردازشی و ذخیره سازی منابع کروموزوم و همچنین مقدار fitness آن به روز می‌شوند.

۳-۶- شرایط توقف الگوریتم

اجرای الگوریتم تا زمانی ادامه می‌یابد که یکی از شرایط خاتمه اتفاق بیافتد. این شرایط عبارتند از :

۱) تعداد تولید نسل ها به حد ماکزیمم برسد.

۲) در صورتی که fitness بهترین راه حل بعد از یک تعداد تکرار خاص تغییر زیادی نداشته باشد.

۳) همگرا شدن fitness کروموزوم ها

۴- نتایج شبیه‌سازی و تحلیل آن‌ها

برای شبیه سازی محیط محاسبات ابری نرم‌افزارهای متعددی مانند کلاد آنلایز، گرین کلاد، ریل کلاد، و... وجود دارد. اما بهترین نرم افزار شبیه سازی محیط محاسبات ابری، شبیه ساز کلادسیم است. شبیه ساز کلادسیم، نرم افزاری کلی است که توسط آن می‌توانیم محیط محاسبات ابری را مدل کنیم و همچنین از آن برای تست کارایی سرویس‌ها کاربردی استفاده می-کنند. این شبیه ساز مبتنی بر زبان جاوا است و یک شبیه ساز مبتنی بر رخداد است. یعنی موجودیت‌هایی که در آن تعریف می‌شوند، از طریق فرستادن رویداد با یکدیگر ارتباط برقرار می‌کنند.

۴-۱- Makespan

همانطور که در جدول (۴) مشاهده می‌کنید الگوریتم پیشنهادی توانسته است زمان تکمیل آخرین کار را تا حدودی کاهش دهد که این بهبود می‌تواند تاثیر بسیار مهمی بر افزایش راندمان سرور ابر داشته باشد.

Archive of SID

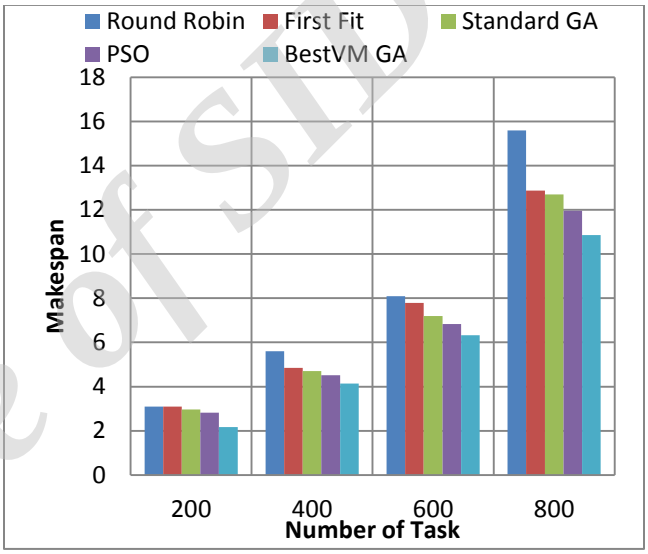
www.SID.ir

number of task	200	400	600	800
Round Robin	3.1	5.6	8.1	15.6
First Fit	3.09	4.85	7.79	12.88
Standard GA	2.96	4.7	7.2	12.7
PSO	2.82	4.52	6.83	11.96
BestVM GA	2.17	4.14	6.32	10.86

جدول (۴) میزان makespan الگوریتم‌ها بر حسب تعداد task ورودی

همانطور که در جدول (۴) مشاهده می‌کنید الگوریتم مورد نظر makespan را نسبت به الگوریتم Round Robin تا حدود ۳۰٪ و نسبت به الگوریتم ژنتیک استاندارد تا حدود ۱۴٪ کاهش داده است.

مقایسه و نحوه رشد الگوریتم‌های ارائه شده در نمودار (۱) مشخص شده است:



نمودار (۱) مقایسه میزانmakespan بین الگوریتم‌های موردنظر

۴-۲-Response time:

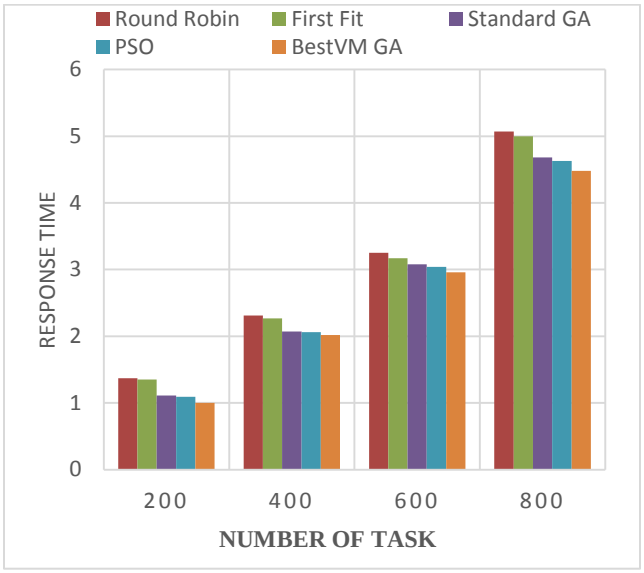
با توجه به جدول (۵) الگوریتم پیشنهادی توانسته‌است زمان تکمیل آخرین کار را تا حدودی کاهش دهد که این بهبود نیز می‌تواند تاثیر بسیار مهمی بر افزایش راندمان سرور ابر داشته باشد:

number of task	200	400	600	800
Round Robin	1.37	2.31	3.25	5.07
First Fit	1.35	2.27	3.17	5
Standard GA	1.11	2.07	3.08	4.68
PSO	1.09	2.06	3.04	4.63
BestVM GA	1	2.02	2.96	4.48

جدول (۵) مقایسه میزان زمان پاسخ میانگین الگوریتم ها

همانطور که در جدول (۵) مشاهده می‌کنید الگوریتم پیشنهادی زمان پاسخ میانگین را نسبت به الگوریتم Round Robin تا حدود 12٪ و نسبت به الگوریتم ژنتیک استاندارد تا حدود 5٪ کاهش داده است.

مقایسه و نحوه رشد الگوریتم‌های ارائه شده در نمودار (۲) مشخص شده است:



نمودار (۲) مقایسه زمان پاسخ میانگین الگوریتم‌ها

۴-۳- نتیجه گیری

تکنولوژی رایانش ابری به صورت گسترده در شرکت‌های تجاری مهم استفاده می‌شود. رایانش ابری بخش‌های بزرگ و کاملی دارد که شامل سخت افزار، نرم افزار و امنیت می‌باشد. رایانش ابری در این سال‌ها یکی از موضوعات تحقیقاتی مهم در زمینه علوم کامپیوتر بوده‌است. این تکنولوژی یک مدل برای دسترسی و استفاده از منابع اشتراکی می‌باشد، که این منابع می‌تواند سرویس، کارهای پردازشی، دخیره‌سازی و .. باشند. در این تحقیق پس از آشنایی با مفاهیم رایانش ابری و بررسی آن، مسائل مربوط به رایانش ابری آشنا شدیم. سپس مسئله مهم تخصیص منابع در رایانش را بررسی کردیم و روش‌های تخصیص منابع در سرورهای رایانش ابری را دسته‌بندی کرده و توضیح دادیم و با مشکلات، مزایا و راهکارهای این روش‌ها آشنا شدیم. سپس از بین روش‌ها و الگوریتم‌های مطالعه شده الگوریتم ژنتیک را جهت بهبود کارایی انتخاب کردیم که دلایل این انتخاب در بخش ۳ توضیح داده شد.

نهایتا یک الگوریتم بر مبنای الگوریتم ژنتیک ارائه کردیم که تا حدودی توانست الگوریتم ژنتیک را در دو فاکتور مهم makespan و response time بهبود دهد. این روش برخلاف الگوریتم ژنتیک استاندارد در مرحله انتخاب نسل اولیه به جای انتخاب تصادفی به صورت کاملا هدفمند نسل اولیه را انتخاب می‌کرد و همچنین در تابع ترکیب نیز تغییراتی ایجاد شد، این موضوع باعث شد الگوریتم مورد نظر با توجه به شرایط محیط سرور ابر کارآیی بالایی از خود نشان دهد.

Archive of SID

www.SID.ir

با توجه به اینکه تکنولوژی رایانش ابری خصوصا در کشور عزیزمان ایران، هنوز به بلوغ کافی نرسیده و نیازمند مطالعه و تلاش بیشتری می‌باشد، امیدواریم با افزایش تحقیقات در این زمینه وتلاش و کوشش اساتید عزیز شاهد رشد روزافزون معلومات دانش‌پژوهان کشورمان در این زمینه باشیم.

رایانش ابری و مخصوصا قسمت مدیریت منابع فضای زیادی برای انجام پروژه‌های تحقیقاتی دارد و می‌توان در این زمینه شاهد کارها و ایده‌های جدید و حتا بهبود دادن ایده‌های قبلی بود.

مراجع

[۱] آرش قربان نیا دلاور، یلدا آرین، ۱۳۹۱، "ارائه یک روش زمان‌بندی ترکیبی بر مبنای الگوریتم ژنتیک در محیط محاسبات ابری"، اولین همایش ملی فناوری اطلاعات و شبکه‌های کامپیوتری دانشگاه پیام‌نور.

[۲] ف.ابراهیمی، افراهی و اکبر فرهودی‌نژاد، ۱۳۹۱، "مروری بر روش‌های تخصیص منابع و اهمیت آن در محیط رایانش ابری"، اولین همایش ملی فناوری اطلاعات و شبکه‌های کامپیوتری دانشگاه پیام‌نور.

[3] A.Heger, 2009, "Optimized Resource Allocation & Task Schedulling Challenges in Cloud Computing Environments", DHTecnologies (DHT), 2009.

[4] A.Kr.Singh, 2012, "A Survey On Resource Allocation Policies in Cloud Computing Environments" in Proce Of Golden Research Thoughts, VOL.2, Issue.4, October 2012, pp.1-8.

[5] Amandeep Verma, Sakshi Kaushal, "Deadline constraint heuristic-based genetic algorithm for workflow scheduling in cloud" , International Journal of Grid and Utility Computing Volume 5 Issue 2, March 2014 .
.

[6] Ch.S.Pawar and R.B.Wngh, 2013, "Priority Based Dynamic Resource Allocation in Cloud Computing with Modified Waiting Queue", International Conference on Intelligent System and Signal Processing, 2013, pp.311-316.

[7] Daniel Oliveira , Kary A. Ocaña , Marta Mattoso , "A Provenance-based Adaptive Scheduling Heuristic for Parallel Scientific Workflows in Clouds" , Journal of Grid Computing , Springer-Verlag New York Volume 10 Issue 3, September 2012 .

[8] Daji Ergu, Yi Peng, Yong Shi, Yu Shi, "The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment" , The Journal of Supercomputing , Volume 64 Issue 3, June 2013 .

[9] Fu.Xie, Y.Du and H.Tina, 2013, "A Resource Allocation Strategy Based on Particle Swarm Algorithm in Vloud Computing Environment", fourth International Conference on Digital manufacturing & Automation, pp.69-72.

[10] F.Wuhib, R.Stadler and H.Lindgron, 2012, "Dynamic Resource Allocation with Management Objectives-Implementation for an OpenStack Cloud", KTH Technical Report TRHA-EE, 2012, pp.1-8.

[11] G.E.Goncalres, P.T.Endo and T.Damancono, 2011, "Resource Allocation in Cloud Concepts, Tools and Research Challenges", Facutly of Computer UFMS, 2011, pp.214-226.

[12] H.Liang, D.Huang, L.X.Cai, X.Shen, and D.peng, 2011, "Resource Allocation for Security Services in Mobile Cloud Computing", Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on Shanghai, April 2011, pp.191-195.

[13] J.Gu, J.Hu, T.Zhao and G.sun, 2012, "A new Resource Scheduling Strategy Based on Genetic Algorithm in Cloud Computing Environment", Journal of Computers, VOL.7, NO.1, January 2012, pp.42-52.

[14] Jinn-Tsong Tsai, Jia-Cen Fang, Jyh-Horng Chou , "Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm" , journal of Computers and Operations Research , Volume 40 Issue 12, December, 2013 .

[15] L.Zhu.Q.Li and L.He, 2012, "Study on Cloud Computing Resource Scheduling Strategy Based on the Ant colony optimization Algorithm", IJCSI International Journal of Computer Seience Issues, VOL.9, Issue.5, NO.2, September 2012, pp.54-58..

[16] M.Ying, Luo.Ke, L.Jian-hua and J.fel, 2012, "A-Rough Sets K-means Resource Dynamiv Allocation Strategy based on Cloud Computing Environment", TELKOMNIKA Indonesian Journal of Electrical Engineering, VOL.10, NO.6, October 2012, pp.1485-1489.

[17] Q.Zhang, Q.Zhu and R.Boutaba, 2011, "Dynamic Resource Allocation for spot Markets in Cloud Computing Environment", Fourth IEEE International Conference, 2011, pp.178-185.

[18] V.Ankina 2013, "A Survey on Various Resource Allocation policies in Cloud Computing Environment", International Journals of Researcher in Engineering and Tecnology (LYRET), ISSN: 2319-1163, VOL.2, Isuue 5, May 2013, pp.760-763.

[19] V.Vinothina, R.Sridaran and P.Ganapathi, 2012, "A Survey on Resource Allocation Strategies in Cloud Computing", International Journals Advanced Computer Science and Applications, 2012, pp.97-104.

[20] Y.Wang, L.Meng and X.Qiu, 2012, "A Multi-dimensional Resource Allocation Algorithm in Cloud Computing", inproce of Journal of Information & Computational, Octobr 2012, pp.3021-3028.

Archive of SID

www.SID.ir