Archive of SID

ارائه مدل ترکیبی زمان‌بندی وظایف براساس الگوریتم تطبیقی پویا و ژنتیک در محیط رایانش ابری

مژده ربانی
دانشگاه آزاد اسلامی، یزد، ایران
mrabbani@iauyazd.ac.ir

هدی مرادی
دانشگاه آزاد اسلامی، یزد، ایران
h.moradi@iauyazd.ac.ir

اسداله علیرضایی*
دانشگاه آزاد اسلامی، یزد، ایران
Alirezaei.emba@gmail.com

تاریخ دریافت: ۱۳۹۸/۰۴/۱۸
تاریخ اصلاحات: ۱۳۹۹/۰۵/۱۹
تاریخ پذیرش: ۱۳۹۹/۰۸/۰۶

چکیده

رایانش ابری مفهومی است که اخیراً در دنیای فناوری اطلاعات معرفی شده که محیطی را برای به اشتراک‌گذاری منابع فراهم می‌نماید. در محیط رایانش ابری یک سیستم زمانبندی کارا و مؤثر، نقش کلیدی دارد و در میزان موفقیت آن محیط بسیار تعیین‌کننده می‌باشد. از این‌رو مسأله زمان‌بندی وظایف در رایانش ابری، مسأله‌ای بسیار مهمی محسوب می‌شود که سعی دارد یک زمان‌بندی بهینه برای اجرای وظایف و تخصیص منبع بهینه مشخص نماید. این پژوهش با هدف ارائه رویکردی جدید برای بهینه‌سازی زمان‌بندی وظایف براساس ترکیبی از الگوریتم تطبیقی پویا و الگوریتم ژنتیک در محیط رایانش ابری انجام گرفته است. مدل ترکیبی، با هدف پوشش‌دادن نقاط ضعف دو الگوریتم ژنتیک و زمان‌بندی تطبیقی پویا، ارائه شد که به موجب آن ضعف الگوریتم ژنتیک یعنی سرعت عمل پایین، با ایجاد یک راه‌حل اولیه مناسب برطرف گردید و همچنین ضعف الگوریتم زمان‌بندی تطبیقی پویا که همانا به دام افتادن در جواب‌های محلی است، با استفاده از الگوریتم بهینه‌سازی تکاملی پوشش داده شد. در مدل پیشنهادی نخست الگوریتم تطبیقی پویا با تکنیک جستجوی منابع مورد نیاز به کمک درخت کی‌دی و بازیابی داده‌ها به صورت موازی توانسته است در زمان بسیار خوبی جواب مناسبی برای مسأله‌ی زمان‌بندی پیدا کند، که به‌عنوان جواب اولیه به الگوریتم ژنتیک سپرده شد و الگوریتم ژنتیک نیز به جستجوی بهینه می‌پردازد. در نتیجه الگوریتم نهایی حاصل از ترکیب این دو، توازن بار بهتری برای منابع ابری در مدت زمان نسبتاً قابل قبول ارائه می‌دهد.

واژگان کلیدی

رایانش ابری؛ زمانبندی؛ الگوریتم تطبیقی پویا؛ الگوریتم ژنتیک؛ درخت کی‌دی.

۱- مقدمه

محاسبات ابری مفهومی است که اخیراً در دنیای فناوری اطلاعات معرفی شده [۱،۲] که ماحصل ترکیب محاسبات گرید، محاسبات توزیع شده و محاسبات سودمند می‌باشد. این فناوری در محاسبات بزرگ و ذخیره داده‌ها به دلیل توانایی به اشتراک‌گذاری منابع محاسباتی که در سراسر جهان، بیشتر و بیشتر محبوب شده است [۲۱]. چرا که محیطی را برای به اشتراک‌گذاری منابع فراهم می‌آورد که استفاده مؤثر از آن می‌تواند منجر به افزایش بهره‌وردی در سیستم شود [۳،۲]. همچنین با توجه به مزایایی که به ارمغان می‌آورد مانند قدرت محاسباتی بالا، هزینه خدمات پایین، عملکرد بهتر، مقیاس‌پذیری و قابلیت دسترسی، موجب شد که رایانش ابری به یک ابزار تبدیل شود. رایانش ابری و در کل رایانش توزیعی، یک مدل شبکه‌ای جدید است که با استفاده از منابع سخت‌افزاری محاسباتی و دیتاسنتر متصل به هم، امکان انجام کارهای سنگین را فراهم می‌کند [۴،۵]. و موجب بهبود بهره‌وری و افزایش مقیاس‌پذیری و صرفه‌جویی در هزینه‌ها و ... می‌گردد [۲]. در محیط ابری به دلیل قابلیت مجازی‌سازی آن، هیچ‌یک از کاربران با جزئیاتی مانند منابع اطلاعاتی، تجهیزات سخت‌افزاری، سیستم عامل و ... که در ابر قرار دارند سر و کار نداشته و به اصطلاح آن‌ها را نمی‌بینند [۱،۲۱]. کاربران فقط درخواست خود را با نرخ‌های ارسالی متفاوت ارسال می‌کنند و سیستم زمان‌بند محیط ابری درخواست را دریافت کرده و با استفاده از الگوریتم زمان‌بند، به اختصاص منابع ابری به پردازش کارهای کاربران می‌پردازد و اطلاعات پردازش شده برای آنان ارسال می‌شود [۲۲،۶]. با افزایش تقاضا، مسلماً راهبردهای زمان‌بندی درخواست‌ها از اهمیت بسیار زیادی برخوردار می‌شود، به صورتیکه واحد زمان‌بندی که شامل الگوریتم زمان‌بندی است، وظایف و برنامه‌های درخواستی را طوری باید برنامه‌ریزی می‌کند که کاربران را خرسند نموده و در عین حال مدیر ابر نیز متضرر نشود به نحوی که توان و بازدهی کل سیستم افزایش یابد [۷]. علاوه بر این در

* نویسنده مسئول

www.SID.ir

Archive of SID

سیستم‌هایی که از زمان‌بندی استفاده می‌کنند ممکن است برخی از کاربران به مدت زمان بیشتری برای انجام کار نیاز داشته باشند و یا ممکن است برخی از کارهای کوچک زودتر از مدت مشخص شده خاتمه یابد. بنابراین به هنگام زمان‌بندی، برنامه‌ریز باید محدودیت‌ها را در نظر بگیرد، از جمله ماهیت کار، اندازه کار، زمان اجرای کار، در دسترس‌بودن منابع، صف کار و ... [۲۲]. پس باید یک مکانیزم زمان‌بندی منابع داده‌ای را انتخاب کرده و در زمانی بهینه به منابع پردازشی ارسال کند. همچنین انتخاب منبع رایانش باید در ارتباط با منابع داده‌ای بوده، ضمن آنکه زمان اجرا نیز کمینه باشد. در این راستا پارامترهای متفاوتی در نظر گرفته می‌شود. بنابراین پیداکردن زمان‌بندی بهینه برای چنین محیطی با استفاده از روش‌های سنتی کار چندان ساده‌ای نیست [۲۳،۲۴]. از طرفی می‌دانیم که موفقیت در محیط رایانش ابری به میزان بسیار زیادی وابسته به الگوریتم به‌کار رفته جهت زمان‌بندی و برنامه‌ریزی درخواست‌های کاربران و مدیریت منابع موجود در ابر است. همچنین وجود الگوریتم‌های مناسب موجب مفید واقع‌شدن ایده‌ی رایانش ابری می‌شود؛ اما با وجود کلیه این فواید، رایانش ابری هنوز هم نیازمند الگوریتم‌های زمان‌بندی بهتر و مؤثرتر از قبل است و دانشمندان حوزه علوم کامپیوتر در تلاش هستند تا به روش‌های جدید دست یابند [۲۵]. تاکنون بسیاری از پژوهشگران الگوریتم‌های اکتشافی و فرااکتشافی زیادی در زمینه زمان‌بندی در محیط‌های ابری پیشنهاد و مورد بررسی قرار داده‌اند که ممکن است منجر به استخراج جواب‌های نزدیک به بهینه شد، اما این روش‌ها دارای مشکلات عمده‌ای از جمله کیفیت سرویس پایین، هزینه اجرای بالا، ضعف در تقسیم بار، ضعف در مدیریت پهنای باند شبکه ابری و عدم پردازش مناسب داده‌ها بوده‌اند. لذا در این پژوهش از مدلی ترکیبی استفاده خواهیم کرد که ضمن مرتفع‌کردن معایب روش‌های مورد استفاده، علاوه بر ارائه زمان‌بندی مناسب برای وظایف، از دقت خوبی نیز برخوردار باشد. در این راستا در الگوریتم پیشنهادی از ویژگی‌های ممتاز دو روش زمان‌بندی یعنی الگوریتم ژنتیک و زمان‌بندی تطبیقی پویا بهره می‌بریم و سعی می‌کنیم تا از معایب دو روش مذکور یعنی اکسترمم‌های محلی و سرعت پاسخگویی پایین تا حد امکان دور شویم.

در مرحله اول جهت زمان‌بندی برنامه‌های ابری با استفاده از الگوریتم تطبیقی پویا، علاوه بر در نظر گرفتن حالت جاری سیستم، وظایف و منابع در ساختارهای درختی قرار می‌گیرند و داده‌های مورد نیاز هر برنامه تکه تکه شده و به صورت موازی به نودهای کامپیوتری منتقل می‌گردد. در این روش برخلاف روش‌های معمول، به جای استفاده از زمان تقریبی اجرای وظایف (میانگین زمان تکمیل پردازش)، از نزدیک‌ترین زمان اجرای ممکن استفاده می‌شود. در نتیجه به دلیل استفاده از ساختمان داده‌های درختی و شبکه‌ای، سرعت بالایی دارد و بنابراین برای محیط‌های ابری با حجم زیاد از وظایف و درخواست‌ها مناسب است [۲۶]. و به افراد این امکان را می‌دهد که به سرعت راه‌حل اولیه مناسبی برای زمان‌بندی وظایف در رایانش ابری دست یابند؛ اما لزوماً خروجی الگوریتم تطبیقی پویا بهترین راه‌حل نخواهد بود. از این‌رو جواب حاصل می‌بایست به الگوریتم ژنتیک سپرده شود تا راه‌حل بهینه‌تر و دقیق‌تر توسط آن جستجو شود. در الگوریتم ژنتیک پارامترهای مسأله بهینه شونده به شکل بردارهایی به نام کروموزوم[1] نگاشت می‌شوند. هر کروموزوم، به‌عنوان یک جواب منفرد مسأله، رشته‌ای از ژن‌هاست که هر ژن متناظر با یکی از پارامترهای مسأله می‌باشد. ابتدا یک جمعیت اولیه از کروموزوم‌ها به‌طور تصادفی تولید می‌شوند. تابع شایستگی[2]، جواب‌های منفرد را ارزیابی کرده و مقدار ارزشمندبودن آنها را محاسبه می‌کند. در گام بعد مکانیزم‌های انتخاب طبیعی شامل برش[3] و جهش[4] بر کروموزوم‌های موجود اثر گذاشته و نسل بعدی جواب‌های منفرد را تولید می‌کند. در نهایت برخی از مناسب‌ترین جواب‌ها با ضریب بیشتر در کنار چند جواب معمولی با ضریب کمتر گزینش[5] می‌شوند و این روند تا رسیدن به معیار توقف و جواب مناسب ادامه می‌یابد.

این پژوهش با توجه به نقش کلیدی و مؤثر زمان‌بندی در محیط رایانش ابری [۲۷] با هدف یافتن رویکرد ترکیبی جدید، از روش‌های زمان‌بندی موجود و به منظور بهینه‌کردن مسأله‌ی زمان‌بندی و کاهش زمان واکنش به درخواست‌های کاربران تحت شبکه انجام شده است. به عبارتی سعی بر آن است که با ایجاد یک مدل ترکیبی و با بهره‌گیری حجم زیادی از وظایف را به کمک روش تطبیقی پویا زمان‌بندی کرده و سپس با کمک الگوریتم ژنتیک به جواب دقیق‌تر دست یابیم. همچنین در قالب هدف کلی فوق اهداف جزیی مانند شناسایی و بررسی یک مدل خوشه‌بندی مناسب مانند k-tree برای الگوریتم تطبیقی پویا جهت تشکیل یک ساختار جستجوی مناسب برای منابع ابری. و همچنین شناسایی و بررسی الگوریتم ژنتیک به‌عنوان یک ابزار قدرتمند و دقیق بهینه‌سازی و تنظیم پارامترهای آن به منظور افزایش دقت زمان‌بندی اشاره کرد. بنابراین، با بهره‌گیری از نقاط قوت دو روش یاد شده، توازن بار بهتری بر روی منابع ابری نسبت به آن دو روش برقرار نماید. ساختار پژوهش حاضر که در ادامه آمده است بدین شرح است: بخش دوم شامل مروری بر متغیرهای پژوهش، بخش سوم مروری بر کارهای پیشین در حوزه زمان‌بندی وظایف در محیط‌های ابری، بخش چهارم روش‌شناسی پژوهش و بخش پنجم یافته‌های پژوهش و در نهایت در بخش پایانی نتیجه‌گیری ارائه خواهد شد.

۲- پیشینهٔ پژوهش

از ابتدای تولد رایانش ابری تا بدین روز همواره در زمینه‌ی زمان‌بندی وظایف، تحقیقات گسترده انجام گرفته و در نتیجه‌ی آنها، روش‌ها و الگوریتم‌های متفاوتی معرفی شده است. بطور کلی الگوریتم‌های زمان‌بندی

1. Chromosome
2. Fitness Function
3. Crossover
4. Mutation
5. Selection

www.SID.ir

Archive of SID

به دو دسته اکتشافی و فرااکتشافی تقسیم می‌شود. هدف از الگوریتم‌های اکتشافی، ارائه راه‌حل در چارچوب یک زمان قابل قبول است که برای حل مسائل مناسب باشد این الگوریتم‌ها نزدیک‌ترین راه‌حل به راه‌حل واقعی را برای مسأله ارائه می‌دهند و در مواردی که مشکل بهینه‌سازی رو به رو هستیم می‌توان از الگوریتم‌های فرااکتشافی بهره گرفت که ترکیبی از الگوریتم‌های اکتشافی است که جهت تولید، انتخاب و یا یافتن هر اکتشاف در هر مرحله طراحی می‌شود. از آنجا که زمان‌بندی از چالش‌های مهم در رایانش ابری است [۷،۸]، و هدف از سیستم‌های محاسبات ابری به حداقل رساندن هزینه و به حداکثر رساندن درآمد حاصل از سرویس‌های ابری است. بنابراین مطالعات و پژوهش‌های زیادی در رابطه با آن انجام گرفته است که گزیده‌ای از آنها در این بخش ارائه می‌گردد.

۲-۱-۱- فرااکتشافی:

مطالعات داخلی:

طهماسبی و خیام‌باشی (۱۳۹۷) به پژوهشی تحت عنوان ارائه مدل رایانش ابری وسایل نقلیه برای بهینه‌سازی هزینه و زمان پاسخگویی پرداختند. آنها مدلی را ارائه دادند که می‌توانست به تخمین هزینه‌های پیاده‌سازی کمک کند. در مدل ارائه شده هزینه انتقال محاسبات با توجه به مکان انجام کار محاسبات، مشخصات سخت‌افزاری منابع محاسباتی و اولویت منابع لحاظ گردید. و با اولویت‌دادن به منبع محاسباتی درون وسیله نقلیه، از هزینه کلی کاسته شد و علاوه بر آن زمان پاسخگویی بهتری برای وظایف سبک بدست آوردند. نتایج نشان داد که مدل ارائه شده می‌توانست دیدگاهی از سیستم به همراه مزایا و معایب نشان دهد [۹].

جمالی و شاکری (۱۳۹۷) به پژوهشی با عنوان زمان‌بندی ایستای وظایف در محیط رایانش ابری با استفاده از الگوریتم ترکیبی بهینه‌سازی ازدحام ذرات و جستجوی ممنوعه با هدف کاهش زمان تکمیل کل کارها پرداختند. در این پژوهش الگوریتم جستجوی ممنوعه به ازای هر ذره از الگوریتم بهینه‌سازی ازدحام ذرات اجرا شد تا از به دام افتادن در بهینه محلی جلوگیری شود و متعاقباً در پویش بیشتر فضای جستجو و دستیابی به جواب‌های بهتر مؤثر واقع شود. نتایج حاکی از برتری الگوریتم ترکیبی پیشنهادی، با توجه به معیار زمان تکمیل کل کارها نسبت به الگوریتم‌های پایه ژنتیک، بهینه‌سازی ازدحام ذرات، جستجوی ممنوعه، الگوریتم خروج به ترتیب ورود و الگوریتم بزرگ‌ترین کوچک‌ترین بود [۱۰].

میسوندی و جعفرزاه (۱۳۹۶) به پژوهشی تحت عنوان ارائه یک الگوریتم جدید جهت ایجاد تعادل بار روی ماشین‌های مجازی در رایانش ابری پرداختند که هدف از آن ارائه یک الگوریتم جدید جهت ایجاد تعادل بار در رایانش ابری با به حداقل رساندن زمان تکمیل آخرین کار و به حداکثر رساندن توان عملیاتی بود نتایج نشان داد که روش پیشنهادی که ترکیبی از الگوریتم بهینه‌سازی ازدحام ذرات و راهبرد جهش الگوریتم ژنتیک است، از کارائی بالایی برخوردار است و می‌تواند به تعادل بار خوبی در محیط رایانش ابری مقیاس بزرگ نسبت به الگوریتم‌های تعادل بار قبلی برسد [۱۱].

رجبی، رشیدا و مردوخی (۱۳۹۵) در پژوهشی تحت عنوان مروری بر الگوریتم‌های فرا اکتشافی در رایانش ابری و ارائه یک الگوریتم هیبریدی برای حل مسأله زمان‌بندی کارها در ابر به بررسی نحوه عملکرد الگوریتم‌های بهینه‌سازی کلونی مورچگان، بهینه‌سازی انبوه ذرات، رقابت استعماری، ژنتیک و شبیه‌سازی تبرید پرداختند و سپس به بررسی هیبرید الگوریتم ژنتیک، که یک نوع ترکیب الگوریتم‌ها است با سایر الگوریتم‌ها پرداختند و در نهایت به این نتیجه رسیدند که هیبرید الگوریتم ژنتیک با الگوریتم بهینه‌سازی انبوه ذرات عملکرد بهتری نسبت به هیبرید سایر الگوریتم‌ها دارد و زمان اتمام اجرای کارها و روند زمانی اجرای کارها را بهبود بخشیدند [۱۲].

همچنین رجبی، رشیدا و مردوخی (۱۳۹۵) در پژوهشی دیگر تحت عنوان ارائه الگوریتم ترکیبی بر مبنای الگوریتم ژنتیک برای زمان‌بندی کارها در رایانش ابری، جهت زمان‌بندی کارهای توزیع شده به ارائه الگوریتم‌های تکاملی جدید که حاصل ترکیب الگوریتم ژنتیک و الگوریتم بهینه‌سازی انبوه ذرات، رقابت استعماری، شبیه‌سازی تبریدی، کلونی مورچگان بود، پرداختند، الگوریتم‌های پیشنهادی عملکرد الگوریتم ژنتیک را بهبود بخشیده و یافتن بهینه سراسری را تضمین می‌کند و با اندازه‌گیری زمان انجام کارها نسبت به الگوریتم‌های ساده هر یک می‌تواند میزان بهبود الگوریتم پیشنهادی را محاسبه نماید و در نهایت الگوریتم ترکیبی پیشنهادی ژنتیک - بهینه‌سازی انبوه ذرات به‌عنوان بهترین الگوریتم برای مسأله زمان‌بندی کارها در سیستم ابری انتخاب شد [۱۳].

حبیبی و همکاران (۱۳۹۳) به پژوهشی تحت عنوان رویکرد جدید برای حل مسأله زمان‌بندی کارها در محیط محاسبات ابری با استفاده از الگوریتم ژنتیک و جستجوی حرام پرداختند. نتایج نشان داد که الگوریتم پیشنهادی از کارایی بالایی برخوردار است و توانسته است یک توازن بین معیارهای کاربران بوجود آورد. منظور از توازن این است که در الگوریتم پیشنهادی سعی شده است که رضایت همه کاربران را جلب کند و همچنین کارها بر روی پردازنده‌های مناسب مورد پردازش قرار گیرد تا کارهای بیشتری در زمان کمتری در محیط ابری مورد پردازش قرار گیرد [۱۴].

مطالعات خارجی :

میلان و همکاران (۲۰۱۹) در پژوهشی تحت عنوان الگوریتم‌های فرااکتشافی با الهام از طبیعت برای حل مسأله توازن بار در محیط‌های ابری پرداختند. در این پژوهش ۱۴ الگوریتم فرااکتشافی (ACO, PSO GA, BCO, FA, CSA, HSA, BA, MA, SOSO, SA, SFLA, Spiderweb). الهام گرفته از طبیعت را در حوزه توازن بار (به حداقل رساندن سربار و زمان پاسخ‌دهی و بهبود عملکرد) مورد بررسی و مقایسه قرار دادند. نتایج به‌دست آمده نشان داد که ACO، GA و BCO الگوریتم‌هایی هستند که موجب بهبود در استفاده بهینه از منابع می‌شوند [۲۸].

www.SID.ir

Archive of SID

مپتو و همکاران (۲۰۱۹) پژوهشی را تحت عنوان الگوریتم بهینه‌سازی ازدحام ذرات باینری برای زمان‌بندی وظایف و توازن بار در محاسبات ابری انجام دادند. آنها از یک نسخه باینری الگوریتم PSO (BPSO) با پیچیدگی و هزینه کم برای زمان‌بندی وظایف و توازن بار در محاسبات ابری و به منظور حداقل‌کردن زمان انتظار، درجه عدم توازن و در عین حال به حداقل رساندن زمان و هزینه اجرا و به حداکثر رساندن استفاده از منابع، معرفی کردند. نتایج نشان داد که الگوریتم پیشنهادی، زمان‌بندی وظایف و توازن بار بهتری از الگوریتم‌های اکتشافی موجود ارائه می‌دهد. [۲۹]

چاودری و همکاران (۲۰۱۸) در پژوهشی تحت عنوان الگوریتم ترکیبی مبتنی بر GSA برای برنامه‌ریزی گردش کار دو هدفه در محاسبات ابری را انجام دادند. آنها یک الگوریتم فرااکتشافی برای زمان‌بندی جریان کار پیشنهاد دادند که مدت زمان و هزینه را در نظر می‌گرفت. این الگوریتم ترکیبی از الگوریتم‌های GSA و HEFT بود. آنها با استفاده از آزمون‌های آماری و آنالیز واریانس اثربخشی الگوریتم پیشنهادی را نسبت به الگوریتم‌های GSA و HEFT نشان دادند به‌طوری‌که نتایج حاکی از آن بود که الگوریتم پیشنهادی از این الگوریتم‌ها بهتر عمل می‌کند [۳۰].

بنالا و همکاران (۲۰۱۸) به پژوهشی تحت عنوان روش برنامه‌ریزی وظایف براساس صف‌های پویا و الگوریتم‌های ترکیبی فرااکتشافی در محیط‌های محاسبات ابری پرداختند. الگوریتم‌های اکتشافی ترکیبی از الگوریتم ازدحام ذرات فازی (TSDQ-FLPSO) و الگوریتم تبرید شبیه‌سازی‌شده ازدحام ذرات (TSDQ-SAPSO) بود. در این پژوهش قصد داشتند زمان انتظار در صف را با استفاده از الگوریتم TSDQ به حداقل برسانند و سپس وظایف را به مناسب‌ترین منابع اختصاص داده و معیارهای عملکرد را بهینه کنند. عملکرد این الگوریتم را با استفاده از شبیه‌ساز CloudSim با الگوریتم‌های MGA SGA, PSO-LDIW, PSO-RIW وMACOLB مورد مقایسه قرار گرفت. نتایج تجربی حاکی از آن بود که الگوریتم‌های TSDQ-FLPSO و SAPSO-TSDQ در به حداقل رساندن زمان انتظار و همچنین طول صف، کاهش عملکرد، کاهش حداکثر استفاده، استفاده از منابع، به حداقل رساندن هزینه اجرای، و بهبود توازن بار، عملکرد خوبی نسبت به سایر الگوریتم‌ها کسب کردند [۳۱].

یونیس و یانگ (۲۰۱۸) به پژوهشی با عنوان ارائه الگوریتم فرااکتشافی برای برنامه‌ریزی شغلی در محیط محاسبات شبکه پرداختند. آنها از دو روش ترکیبی فرااکتشافی برای برنامه‌ریزی شغلی استفاده کردند به نحوی که در مرحله اول ترکیبی از الگوریتم‌های کلونی مورچگان و جستجوی همسایگی متغیر استفاده کردند و در مرحله دوم الگوریتم ژنتیک را با جستجوی همسایگی متغیر، ادغام کردند. آزمایش‌های متعددی برای تجزیه و تحلیل عملکرد انجام شد. نتایج نشان داد که الگوریتم‌های جدید ویژگی‌های روش‌های مورد استفاده را به ارث برده، با این حال الگوریتم ACO-VNS در مقایسه با االگوریتم GA-VNS به زمان بیشتری نیاز دارد [۳۲].

۲-۱-۲- اکتشافی:

مطالعات داخلی:

رمدانی و شیروانی (۱۳۹۷) در پژوهشی تحت عنوان الگوریتم زمان‌بندی ظایف مبتنی بر ژنتیک در سیستم‌های توزیعی ناهمگن پرداختند. آنها یک الگوریتم زمان‌بندی کار مبتنی بر ژنتیک را برای به حداقل رساندن زمان اجرای کل کارها توسعه دادند. در این روش از مزایای روش‌های اکتشافی، برای ایجاد جمعیت اولیه استفاده شد. همچنین اپراتورهای ژنتیکی به نحو مناسب برای تعیین راه‌حل مطلوب نهایی به‌کار گرفته شد. برای رسیدن به نتایج دقیق، چندین سناریو در نظر گرفتند که نتایج نشان داد که الگوریتم پیشنهادی در مقایسه با روش‌های موجود مثل HEFT و QGARAR از لحاظ متوسط زمان اجرا برتری دارد [۱۵].

ابراهیمی هردورودی و همکاران (۱۳۹۷) به پژوهشی با عنوان ارایه روش بهینه‌سازی زمان‌بندی وظایف در رایانش ابری با الگوریتم گرگ خاکستری پرداختند. مراحل اصلی الگوریتم پیشنهادی شامل ردیابی، تعقیب و نزدیک‌شدن به طعمه، تعقیب، محاصره و آزار طعمه تا زمانی‌که از حرکت بازایستد و حمله به سمت طعمه بود. برای اعتبارسنجی روش پیشنهادی از شبیه‌سازی در نرم‌افزارمتلب استفاده شد. نتایج آزمایشات و مقایسه با روش‌های دیگر نشان داد روش پیشنهادی در همگرایی، مصرف پهنای باند و زمان پاسخ بهترین کارایی را نسبت به روش‌های مشابه دیگر داشت [۱۶].

غریبی و جعفری (۱۳۹۶). به پژوهشی تحت عنوان زمان‌بندی اولویت‌دار دارای مهلت زمانی در رایانش ابری به کمک الگوریتم بهینه‌سازی گرگ خاکستری پرداختند و یک روش زمان‌بندی با استفاده از الگوریتم گرگ خاکستری با در نظرگرفتن اولویت و مهلت زمانی هر وظیفه و تخصیص نزدیک‌ترین ماشین مجازی به وظیفه مورد نظر صورت گرفته که مصرف انرژی و زمان اجرای کل وظایف را به حداقل رساندند. نتایج شبیه‌سازی نشان داد که الگوریتم پیشنهادی بهتر از الگوریتم بهینه کلونی مورچگان عمل نموده و میزان مصرف انرژی را ۱۸ درصد و زمان اجرای کل وظایف را تا ۴ درصد بهبود بخشید [۷].

اردستانی و بدیع (۱۳۹۵) در پژوهشی تحت عنوان زمان‌بندی در محیط محاسبات ابری با استفاده از الگوریتم‌های تکاملی، با استفاده از الگوریتم بهینه‌سازی علف‌های هرز زمان‌بندی وظایف در محیط ابر در جهت کمینه‌سازی زمان تکمیل وظایف انجام شده است. در جهت بهبود عملکرد الگوریتم بهینه‌سازی علف‌های هرز از عملگرهای تعویض و وارون‌سازی استفاده کردند تا تنوعی بیشتری در فضای پاسخ ایجاد شود و در صورت افتادن در نقاط بهینه محلی باعث خروج و یافتن پاسخ بهینه سراسری شوند. نتایج آن با الگوریتم ژنتیک مقایسه شد و نتایج حاکی از عملکرد بهینه الگوریتم بهینه‌سازی علف‌های هرز در برابر الگوریتم ژنتیک بود [۱۷].

کهمره و مصلی نژاد (۱۳۹۴) به پژوهشی تحت عنوان ارائه الگوریتم زمان‌بندی در رایانش ابری مبتنی بر هوش مصنوعی پرداختند. آنها از الگوریتم زمان‌بندی قورباغه جهت کاهش زمان‌بندی انجام کارها بهره

www.SID.ir

Archive of SID

گرفتند و نتایج آن را با الگوریتم‌های Aco و Coa مقایسه کردند. نتایج حاکی از آن بود که روش پیشنهادی به لحاظ هوشمندی و دقت زمان‌بندی، از روش‌های مورد بررسی، بهتر عمل می‌کند و همچنین الگوریتم پیشنهادی از قابلیت و همگرایی بالایی برخوردار است [۱۸].

امینی و فاطری (۱۳۹۳) به پژوهشی تحت عنوان روش بهبودیافته زمان‌بندی منابع در رایانش ابری با استفاده از برنامه‌نویسی ژنتیک پرداختند. در این پژوهش با توجه به راهبردها برای عملگرهای گزینش، تقاطع و ترکیب تلاش شد یک مدل بهینه برای کمینه‌سازی زمان اجرا و زمان انتظار مطرح شود. به منظور ارزیابی و میزان سنجش و دقت روش آزمایشاتی برای کارایی الگوریتم ژنتیک، میزان متوسط بار سیستم‌ها و نرخ مهاجرت انجام گرفت که نتایج حاکی از آن بود که روش پیشنهادی کارایی و بهره‌وری مناسبی دارد [۱۹].

مطالعات خارجی:

ولیانگری و همکاران (۲۰۲۰) به پژوهشی تحت عنوان جستجوی اکتشافی با الگوریتم ژنتیک برای زمان‌بندی وظایف در محاسبات ابری پرداختند. آنها در این پژوهش یک روش اکتشافی ترکیبی (HSGA) برای بهبود زمان‌بندی با در نظر گرفتن پارامترهایی مانند زمان، توازن بار، استفاده بهینه از منابع را پیشنهاد دادند. در ابتدا این الگوریتم به اولویت‌بندی وظایف با توجه به تأثیر آن‌ها بر دیگران پرداخت. سپس، برای ایجاد یک جمعیت اولیه بهینه و اعمال برخی عملیات مناسب مانند جهش در کنترل و هدایت الگوریتم به یک راه‌حل بهینه، از روش Best-Fit و Round Robin استفاده می‌شود. این الگوریتم با در نظر گرفتن پارامترهای مؤثر در محیط ابر، جواب بهینه را ارزیابی می‌کرد. نتایج حاکی از آن بود که الگوریتم پیشنهادی نتایج بهتری را با تعداد بیشتری وظایف ارائه می‌دهد [۳۳].

سعیدی و همکاران (۲۰۲۰) به پژوهشی با عنوان توسعه مدل چند هدفه بهینه‌سازی الگوریتم ذرات به منظور زمان‌بندی جریان کار در محیط‌های ابری پرداختند. آنها یک رویکرد به منظور حل مسأله زمان‌بندی جریان کار با استفاده از الگوریتم بهینه‌سازی ذرات با در نظر گرفتن چهار هدف متضاد یعنی حداکثر رساندن قابلیت اطمینان و به حداقل رساندن هزینه و مدت زمان انتظار و مصرف انرژی، پیشنهاد دادند. نتایج نشان داد که رویکرد پیشنهادی در مقایسه با الگوریتم‌های LEAF و EMS-C، عملکرد بهتری را به همراه داشت [۳۴].

گاوالی و شینده (۲۰۱۸) پژوهشی تحت عنوان برنامه‌ریزی وظایف و تخصیص منابع در محاسبات ابری با استفاده از الگوریتم اکتشافی انجام دادند. در این پژوهش یک رویکرد اکتشافی ترکیبی از MAHP، BATS+BAR LEPT و الگوریتم تقسیم و حل برای برنامه‌ریزی وظایف و تخصیص منابع بود. در این روش هر وظیفه قبل از تخصیص واقعی آن به منابع ابر با استفاده از فرایند MAHP، پردازش شد و منابع با استفاده از روش بهینه‌سازی BATS +BAR تخصیص داده شدند به صورتی‌که پهنای باند را محدودیت در نظر می‌گرفت. علاوه بر این سیستم پیشنهادی با استفاده از LEPT وظیفه اولویت‌بندی منابع هزینه را بر عهده داشت به‌طوری‌که رویکرد تقسیم و حل سیستم پیشنهادی را بهبود می‌بخشید [۳۵].

مدنی و همکاران (۲۰۱۷) در پژوهشی تحت عنوان مقایسه عملکرد الگوریتم‌های اکتشافی برای برنامه‌ریزی وظایف در محیط محاسبات ابری، شش الگوریتم اکتشافی (FCFS, MCT, MIN/MAX-MIN- MET Sufferage). با هدف مقایسه عملکرد آنها از نظر هزینه، درجه عدم تعادل، سازگاری و توان در سیستم محاسبات ابری با کمک شبیه‌ساز CloudSim مورد مقایسه قرار دادند نتایج نشان داد که الگوریتم کلی MIN-MIN نسبت به سایر الگوریتم‌ها عملکرد بهتری دارد. درحالی‌که الگوریتم MAX-MIN و Sufferage نتایج خوبی را ارائه دادند و الگوریتم MET همواره عملکرد بهتری را در دستیابی به میزان عدم تعادل برای برنامه‌ریزی بهینه به همراه دارد [۲۲].

شارما و همکاران (۲۰۱۷) در پژوهشی با عنوان بررسی رویکرد اکتشافی برای زمان‌بندی وظایف در رایانش ابری پرداختند. آنها ۹ الگوریتم اکتشافی را برای زمان‌بندی وظایف مورد بررسی قرار دادند در این پژوهش تجزیه و تحلیل‌های مقایسه‌ای براساس پارامترهای زمان‌بندی، ابزارهای شبیه‌سازی و دامنه مشاهدات و محدودیت‌ها انجام شد. نتایج نشان داد که رویکرد اکتشافی وجود ندارد که بتواند کلیه پارامترهای مورد نیاز را برآورده کند، اما اگر پارامتر خاصی مانند مدت زمان انتظار، استفاده از منابع، زمان اجرای هر کار و گردش کار و ... در یک زمان در نظر گرفته شود، می‌توان انتظار عملکرد بهتری داشت [۳۶].

۳- روش‌شناسی پژوهش

پژوهش حاضر براساس هدف از نوع کاربردی و براساس نحوه شبیه‌سازی محیط ابری، منابع داده و پردازش و وظایف می‌باشد. با توجه به اینکه شبیه‌سازی محیط رایانش ابری در سیستم‌های رایانه‌ای امکان‌پذیر است، داده‌های مورد نیاز پژوهش در محیط شبیه‌سازی کامپیوتری گردآوری شد. عمده‌ی داده‌ها و ابزار لازم عبارتند از یک شبکه‌ی کامپیوتری و اطلاعاتی که بین سیستم‌های موجود در شبکه رد و بدل می‌شوند. در این راستا ابتدا به معرفی ساختار کلی مدل زمان‌بندی وظایف با استفاده از الگوریتم تطبیقی پویا و سپس با استفاده از الگوریتم ژنتیک و در نهایت با استفاده از الگوریتم پیشنهادی (تطبیقی- ژنتیک) پرداخته می‌شود. و در نهایت به بررسی این موضوع می‌پردازیم که آیا با بهره‌گرفتن از ویژگی ممتاز مدل‌ها، می‌توان به یک روش زمان‌بندی مناسب و کارا دست یافت.

همچنین جهت تجزیه و تحلیل داده‌های محیط ابری مورد آزمایش، از نرم‌افزار متلب و سی‌شارپ استفاده گردید و در نهایت نتایج حاصل با استفاده از نمودار و گراف گزارش خواهد شد. لازم به ذکر است که متغیرهای مورد نیاز و چگونگی بررسی و اندازه‌گیری آنها به شرح زیر است:

www.SID.ir

Archive of SID

با فرض اینکه $T_1,\ldots,T_n$ وظایفی هستند که به یک محیط ابری درخواست داده شده و باید توسط ماشین‌های $M_1,\ldots,M_p$ اجرا شوند. همچنین وظایف مذکور به منابع داده‌ای $R_1,\ldots,R_m$ نیاز دارند. هر کدام از وظایف حجم مشخصی داشته و اجرای آنها در زمان مشخصی تکمیل می‌گردد. الگوریتم زمان‌بندی A باید ماشین‌های M_p و منابع داده‌ی R_j را طوری به T_lها اختصاص و اولویت‌بندی نماید تا کم‌ترین زمان ممکن صرف اجرای وظایف گردد. که گام‌های این روند (بررسی و اندازه‌گیری متغیرها) به صورت زیر می‌باشد:

- ایجاد درخت جستجوی ماشین‌ها M_p و منابع R_j
- انتخاب n ماشین با بیشترین توان پردازشی برای هر وظیفه T_i.
- برای هر ماشین M_p منابع همسایه آن به کمک درخت پیدا می‌شود.
- داده‌های مورد نیاز T_i بنا به پهنای باند همسایه‌ها به فایل‌های کوچک‌تر شکسته می‌شود.
- محاسبه‌ی زمان انتقال داده‌ها از نودهای همسایه.
- تخمین زمان شروع به کار وظیفه T_i با توجه به مدت زمان انتقال دیتا و زودترین زمان خالی‌شدن ماشین M_p.
- ماشین M_p ای که کم‌ترین زمان شروع را ممکن می‌سازد به‌عنوان پردازش‌کننده برنامه T_i انتخاب می‌شود.
- گنجاندن جواب به‌دست آمده در جمعیت اولیه الگوریتم ژنتیک.
- جستجوی جواب بهینه توسط الگوریتم ژنتیک.

در ادامه توضیح مختصری از بدنه‌ی اصلی الگوریتم‌های زمان‌بندی تطبیقی پویا و ژنتیک و همچنین روش کار آنها را بررسی نموده با مقایسه‌ی این دو روش برتری هر یک بر دیگری را خواهیم یافت و در پایان ما با تلفیق دو الگوریتم تطبیقی پویا و ژنتیک سعی می‌کنیم جواب بهتری نسبت به دو روش مذکور بیابیم.

۳-۱- الگوریتم زمان‌بندی تطبیقی پویا

ساختار این مدل براساس روش بازیابی اطلاعات مورد نیاز از چندین منبع داده‌ای بوده و با ارایه یک زمان‌بندی پویا و اکتشافی، منابع پردازشگر را به وظایف وابسته به هم براساس زمان بازیابی موازی از چند منبع و زمان پردازش کار اختصاص می‌دهد. بنابراین با ترکیبی از بازیابی داده و تکنیک نگاشت وظیفه - پردازشگر، مدل تطبیقی پویا می‌تواند به زمان‌بندی مناسب و کاراتر نسبت به روش‌های دیگر دست یابد. یک مکانیسم زمان‌بندی باید قادر به انتخاب منابع داده و انتقال اطلاعات از آنها به منابع پردازشگر در زمانی بهینه باشد. از طرفی کامپیوتر اجراکننده وظایف مرتبط با منابع داده‌ای مورد نظر باید طوری انتخاب شود که زمان پردازش کمینه گردد.

به منظور مرتب‌سازی منابع در مدل تطبیقی پویا، منابع داده‌ای و پردازشی در یک ساختار حلقه ای تو در تو دائماً در حال استفاده هستند و الگوریتم زمان‌بندی در میان آنها به جستجو می‌پردازد. بنابراین بسیار ضروری به نظر می‌رسد تا منابع موجود در ابر به صورتی مناسب و کارا دسته‌بندی شوند تا عمل جستجو با کمترین زمان ممکن انجام شود. مشخصاً برای یک سیستم کامپیوتری و دیتاسنتر ملاک و معیارهای چندگانه‌ای برای دسته‌بندی وجود دارد، مانند سرعت سی‌پی‌یو، رَم، درایور هارد دیسک، قدرت و سرعت انتقال داده‌ها در شبکه (پهنای باند) و ... . با توجه به اینکه مرتب‌سازی در این حالت مانند مرتب‌سازی آرایه‌های یک بعدی نیست، مناسب‌ترین گزینه، درخت کی‌دی[1] می‌باشد.

منطق و ساختار اکتشافی الگوریتم تطبیقی پویا به این شرح است: الگوریتم تطبیقی پویا برای هر یک از وظایف جدید وارد شده در صف انتظار، منابع محاسباتی و دیتاسنترها را کاوش کرده و زمان تکمیل وظیفه را بدست می‌آورد. الگوریتم برای کشف سریع‌تر نودهای محاسباتی بیکار یا کم‌کار، همه‌ی آنها را در یک ساختار درختی قرار می‌دهد. زیرمجموعه‌ی nتایی $R=\{R_1,R_2,...,R_n\}$ از منابع پردازشی با بیشترین درجه پیوند بین یکدیگر و بیشترین زمان خالی سی‌پی‌یو از درخت کی‌دی گرفته می‌شود. به یاد داشته باشید درصورتی‌که وظایف تقریباً هم اندازه باشند، زمان بیکاری سی‌پی‌یو را می‌توان با میزان فضای آزاد رَم هم‌ارز در نظر گرفت از آنجا که هرچه فضای رَم آزاد، بیشتری در دسترس باشد به منزله‌ی بارگیری برنامه‌های کمتر و اِشغال کمتر سی‌پی‌یو می‌باشد.

عناصر مجموعه‌ی R منابع کاندید برای اجرای وظایف جدید هستند. برای هر وظیفه i حاضر در صف انتظار، زودترین زمان ممکن برای اجرا توسط نودهای پردازشی R_i محاسبه می‌گردد و زودترین زمان اجرا با در نظر گرفتن زمان پایان اجرای برنامه‌های فعلی بارگیری شده در رَم به دست می‌آید.

Start Time متغیری است که زمان شروع اجرای وظایف[2] را ذخیره کرده و در ابتدا برابر یک مقدار زیاد $(StartTime=\infty)$ در نظر گرفته می‌شود. پس از آن باید مدت زمانی که طول می‌کشد[3] تا داده‌ها و اطلاعات مورد نیاز وظایف به میزبان پردازشگر انتقال یابد، محاسبه شود. از آنجا که این انتقال باید به صورت موازی انجام شود، فایل اطلاعات ورودی به تناسب پهنای باند لحظه‌ای موجود بین نود پردازشگر و دیتاسنترهای همسایه شکسته شده و هر تکه فایل به صورت همزمان از نودهای داده‌ای همسایه وارد میزبان پردازشگر می‌شود. پهنای باند میان سیستم میزبان محاسباتی و دیتاسنتر همسایه برابر با مقدار بوسیله شکل (۱) در نظرگرفته می‌شود:

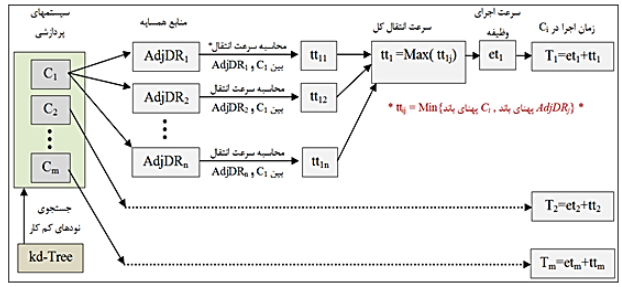
$$\text{BandWidth} = \text{Min}\left\{\begin{array}{c}\text{NetSpeed(CompNode)}\\ ,\\ \text{NetSpeed(AdjacentNode)}\end{array}\right\} \qquad (۱)$$

1. Kd-Tree
2. Task
3. TransferTime

www.SID.ir

Archive of SID

در شکل ۱ نحوه‌ی شکستن فایل اطلاعات نشان داده شده است.



شکل ۱- روند یافتن مناسب‌ترین نود پردازشگر برای یک وظیفه خاص

پیش از اجرای وظایف در میزبان محاسباتی، اطلاعات مورد نیاز آنها باید به نود اجراکننده‌ی وظایف رسیده باشند. بنابراین با مشخص‌شدن زمان انتقال داده‌ها، زمان آغاز به کار وظایف به صورت رابطه (۲) محاسبه می‌گردد.

$$StartTime = EST + TransferTime \quad (۲)$$

هر کدام از نودهای پردازشی R_i که کمترین زمان شروع را داشته باشد، میزبان اجراکننده‌ی وظایف خواهد بود.

۳-۲- زمان‌بندی با الگوریتم ژنتیک

اساس کار زمان‌بندی وظایف در محیط ابری به کمک الگوریتم ژنتیک بسیار ساده و قابل درک است؛ این روند در ۴ گام کلی زیر خلاصه می‌شود: ۱- تبدیل جواب‌های اولیه به کروموزوم[1] ۲- یک یا چند جواب و حدس اولیه۲ ۳- تعیین تابع شایستگی[3] ۴- چرخه تکامل و انتخاب طبیعی.

۳-۲-۱- تبدیل جواب‌های اولیه به کروموزوم

در واقع این قسمت سخت‌ترین بخش کار است چرا که تعیین یک تابع شایستگی مناسب نیازمند تعریف دقیق کروموزم می‌باشد و همچنین سایر قسمت‌ها کلی بوده و بخشی از الگوریتم ژنتیک هستند. این بخش‌ها همه جا و به صورت از قبل آماده وجود داشته و به صورت خودکار اجرا می‌شود و جواب بهینه که یک کروموزوم است را بر می‌گردانند. کروموزوم زبان ارتباط با الگوریتم ژنتیک بوده و جهت اجرای مسأله‌ی زمان‌بندی وظایف در محیط ابری به آن نیازمندیم تا پیش‌نیازها و متغیرهای مسأله را به کروموزوم تبدیل نماییم.

۳-۲-۲- یک یا چند جواب و حدس اولیه:

پس از مدل‌سازی مسأله به زبان ژنتیک، به یک حدس اولیه نیازمندیم؛ تولید این جواب اولیه را می‌توان به عهده‌ی خود الگوریتم ژنتیک گذاشت تا به صورت تصادفی کروموزومی را بسازد. اگر حدس اولیه‌ای داشته باشیم پس از تبدیل به کروموزوم آن را به الگوریتم می‌سپاریم. ایجاد جواب‌های اولیه کاری بسیار ساده است، کافی است ما در ابتدای ورود هر وظیفه، آن را به یک میزبان پردازشی بیکار انتساب دهیم. اما یک نکته‌ی مهم و راهبردی در کار با الگوریتم ژنتیک و دیگر روش‌های تکاملی، تعیین جواب‌های اولیه‌ی خوب می‌باشد. وجود حدس اولیه هم بسیار مفید است ولی هر چه تعداد آنها بیشتر باشد، رسیدن به جواب بهینه سریع‌تر ممکن می‌گردد.

۳-۲-۳- تعیین تابع شایستگی

اصلی‌ترین چالش در بهینه‌سازی مسأله زمان‌بندی وظایف ابری با الگوریتم ژنتیک تعیین تابع شایستگی مناسب بوده که نیازمند تعریف دقیق و کارا از کروموزوم است. تابع شایستگی باید تعدادی کروموزوم حاوی جواب‌های کاندید را دریافت کرده و مقدار تناسب آنها با نیازهای ما را بسنجد. بنابراین اگر کروموزوم برخی اطلاعات را نداشته باشد، تابع شایستگی نمی‌تواند میزان خوب‌بودن آن و در نتیجه جواب مسأله را به درستی اندازه‌گیری کند. درصورتیکه همه‌ی اطلاعات لازم موجود باشد، تابع شایستگی معمولاً به کمک اعمال و توابع ساده ریاضی، ارتباط بین ژن‌های[4] کروموزوم را محاسبه کرده و در نهایت یک مقدار حقیقی به‌عنوان میزان تناسب بر می‌گرداند. با فرض اینکه تعداد سیستم‌های بیکار با تعداد وظایف در حال انتظار برابر است، در این پژوهش همانند پژوهش [۱۰]. با استفاده از تابع شایستگی جایگشت‌های مختلف حاصل از انتساب وظایف به سیستم‌های محاسباتی را امتحان نماییم. بنابراین نیازمند یک تعریف جدید از کروموزوم به صورت رابطه ۳ هستیم.

$$Chromosome=(CNoter_1,CNote_2,\ldots,CNote_N) \quad (۳)$$

اگر اطلاعات مربوط به منابع ابری را در آرایه‌ای به نام ریسورس آرای[5] ذخیره کرده باشیم، متغیرهای $CNote_i$ حاوی اندیس هر یک از کامپیوترهای محاسباتی موجود در آرایه ریسورس آرای هستند. بطور مثال، کروموزوم C_1=(۵،۸،۱۲،۲،۳۵) حاوی اندیس مربوط به اطلاعات پنجمین کامپیوتر، هشتمین کامپیوتر و ... می‌باشد. تکنیک فوق نوعی نگاشت[6] از منابع به وظایف است که به صورت جدول ۱ می‌توان آن را شرح داد [۱۰]:

جدول ۱- نگاشت منابع به وظایف

Task[1]	Task[2]	…	Task[N]
↓	↓	…	↓
CNode[i5]	CNode[i8]	…	CNode[iN]

ترتیب قرار گرفتنها $CNote_i$ به معنای اختصاص داده‌شدن کامپیوتر متناظر آنها به وظایف متناظر با همان شماره جای‌گیری در کروموزوم است. به‌عنوان مثال، کروموزوم C_1 بالا می‌گوید Task1 در سیستم ۵، Task2 در سیستم ۸، ... و Task5 در سیستم ۳۵، اجرا می‌شود. اکنون میزان شایستگی کروموزوم‌هایی به شکل C_1 را با استفاده از تابع حقیقی (۴) محاسبه می‌شود:

$$e = \sum_{i=1}^{N} TransferTime(C_1[i]) + ExecuseTime(C_1[i]) \quad (۴)$$

1. Chromosome
2. Initial Population
3. Fitness Function
4. Genome
5. Resources Array
6. Mapping

www.SID.ir

Archive of SID

به نحوی که:

$$C_1 = (5,8,12,20,35)$$

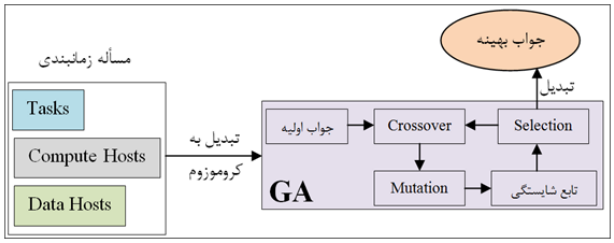
که در آن:

$TransferTime(C_1[i])$: مدت زمان لازم برای انتقال داده‌های مورد نیاز، Task[i] از منابع همسایه به $(C_1[i])$.

$ExecuseTime(C_1[i])$: مدت زمان اجرای Task[i] توسط $(C_1[i])$.

۳-۲-۴- چرخه تکامل و انتخاب طبیعی

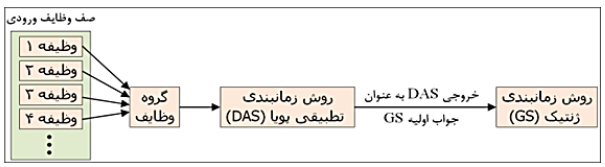
با مشخص‌شدن وضعیت کروموزوم و تابع شایستگی، فرایند تولید نسل‌های جدید توسط سه عمل اصلی الگوریتم ژنتیک: جهش[1]، برش[2] و انتخاب[3] انجام می‌شود. برخی پارامترهای عملیات اصلی الگوریتم مانند ضریب جهش یافتگی، ضریب و نوع برش‌زدن، ضریب نخبه‌گرایی، حجم جمعیت اولیه و ... را می‌توان به دلخواه تغییر داد تا روند رسیدن به جواب بهینه بهبود یابد. در پایان جواب بهینه به دست آمده را به مسأله‌ی اصلی بر می‌گردانیم. شکل ۲ روند کلی زمان‌بندی با الگوریتم ژنتیک را نشان می‌دهد.



شکل ۲- بهینه‌سازی مسأله‌ی زمان‌بندی وظایف محیط رایانش ابری با الگوریتم ژنتیک

۳-۳- مدل پیشنهادی (تطبیقی- ژنتیک)

در مدل پیشنهادی، دو الگوریتم ژنتیک و تطبیقی پویا به‌کار گرفته شده‌اند، تا همزمان از مزایای دو الگوریتم ژنتیک و تطبیقی پویا بهره گرفته شود. در مدل تطبیقی- ژنتیک، به دلیل ماهیت و محدودیت الگوریتم ژنتیک از تکنیک زمان‌بندی دسته‌ای[4] استفاده شد؛ به این معنا وظایف ورودی پس از رسیدن به تعداد مشخصی، زمان‌بندی خواهند شد. در شکل ۳ شمای کلی این مدل نشان داده شده است.



شکل ۳- نمای کلی از الگوریتم تلفیقی ژنتیک - تطبیقی پویا

فرض می‌کنیم سیستم ابری دارای N منبع پردازشی و M منبع داده است که در آن:

مجموعه تمام نودهای پردازشی موجود:

$$C = \{CompNode_1, CompNode_2, ..., CompNode_n\}$$

مجموعه تمام نودهای داده‌ای موجود:

$$D = \{DataNode_1, DataNode_2, ..., DataNode_m\}$$

به منظور سادگی نوشتار، برخی ساختمان داده‌های لازم برای پیاده‌سازی الگوریتم مورد نظر را در ادامه تعریف می‌نماییم.

ساختار ComputeNode: اطلاعات اساسی مربوط به هر میزبان پردازشگر بوده و شامل توان و سرعت سی‌پی‌یو، حجم حافظه‌ی اصلی، سرعت کارت شبکه و لیست وظایفی که میزبان باید آنها را اجرا کند، می‌باشد. البته یادآور می‌شود که در این پژوهش هر میزبان فقط یک وظیفه را اجرا خواهد کرد. بنابراین آرایه ComputeNode.tasks همواره به طول ۱ خواهد بود.

ساختار DataNode: اطلاعات اساسی مربوط به هر میزبان دیتا بوده و شامل حجم HDD، حجم حافظه‌ی اصلی، سرعت کارت شبکه و لیست وظایفی که میزبان داده‌های آنها را ذخیره کرده است، می‌باشد.

ساختار Task: اطلاعات اساسی مربوط به وظایف را نگهداری می‌کند، اطلاعاتی مانند زمان شروع به کار، مدت زمان لازم برای تکمیل کار وحجم برنامه. Task.parent و Task.childs فیلدهای مربوط برنامه‌های وابسته به هم بوده و به دلیل فرض مستقل بودن وظایف، از آنها استفاده نمی‌شود. آرایه‌ی Task.compNodes لیست میزبان‌های اجراکننده را مشخص می‌کند که طول آن همواره ۱ می‌باشد. همچنین آرایه‌ی Task.dataNodes لیست میزبان‌های حاوی داده‌های مورد نیاز وظیفه را نگه می‌دارد. Task.splitedData یک ساختمان داده‌ی خاص است که اطلاعات مربوط به تکه‌های فایل شکسته شده را ذخیره می‌کند. طی روند اجرای الگوریتم تطبیقی پویا، برای محاسبه‌ی سرعت انتقال فایل داده‌های مورد نیاز برنامه‌ی اجرا شونده در میزبان پردازشگر، این فایل به چند فایل کوچک‌تر شکسته شده و مشخصات هر یک در Spliter.splitRatio و Spliter.hostIndex ذخیره می‌شود. Spliter.hostIndex دربردارنده‌ی اندیس منبع داده‌ای است که تکه فایل باید از آنجا دانلود شود و Spliter.splitRatio نسبت اندازه‌ی تکه فایل به اندازه‌ی فایل اصلی بوده و متناسب با پهنای باند موجود بین منبع دیتا و میزبان اجراکننده‌ی وظیفه محاسبه می‌شود.

منابع محاسباتی براساس سه ویژگی (سرعت سی‌پی‌یو، میزان رَم و پهنای باند) در درخت kd-tree به نام TC و نودهای داده‌ای براساس سه ویژگی (حجم هارد، میزان رَم و پهنای باند) در درخت kd-tree به نام TR قرار می‌گیرند. یک درخت kd-tree دیگر با نام T کلیه‌ی منابع محاسباتی و داده‌ای را همزمان و براساس میزان رَم و پهنای باند دسته‌بندی می‌کند.

1. Mutation
2. Crossover
3. Selection
4. Batch Mode

www.SID.ir

Archive of SID

همانطور که گفته شد در مدل پیشنهادی با استفاده از الگوریتم زمان‌بندی تطبیقی پویا جواب اولیه‌ی مناسبی را پیدا کرده که از آن به‌عنوان جواب اولیه‌ی الگوریتم ژنتیک استفاده شد. به دلیل محدودیت کار با الگوریتم ژنتیک، تعداد وظایف و سیستم‌های پردازشگر اجراکننده را یکسان در نظر می‌گیریم. هر زمان که صف انتظار تعداد مشخصی از وظایف ورودی را در خود جای داد، زمان‌بندی مدل پیشنهادی از درخت کی‌دی به همان تعداد منبع پردازشگر با بیش‌ترین امتیاز از نظر پهنای باند و فضای آزاد رَم و سرعت پردازش استخراج کرده و روند اکتشافی فاز تطبیقی پویا آغاز می‌شود. با پایان یافتن این فاز، جواب به‌دست آمده به الگوریتم ژنتیک فرستاده شده و پس از اجرای فاز عملیات بهینه‌سازی، جواب بهینه نهایی دریافت می‌شود. در پایان کروموزوم بهینه به جواب اصلی برگردان می‌شود. قطعه‌ای از قطعه زیر خلاصه‌ای از عملیات الگوریتم تلفیقی را نشان می‌دهد.

فاز یک: الگوریتم تطبیقی پویا

TC: ساختار درخت kd-tree که براساس مشخصات منابع ابری (مانند پهنای باند انتقال داده، قدرت پردازش) ساخته می‌شود.

زیرمجموعه‌ی n تایی $R = (R_1, R_2, \dots)$ از منابع پردازشی با بیشترین درجه پیوند بین یکدیگر از درخت TC گرفته می‌شود.

برای هر وظیفه (Task) به نام $Task_i$ که بهزمان شروع به اجرای Task مقداری یک اضافه می‌شود.

به ازای هر منبع محاسبات R_j در R، همه نودهای همسایه‌ی R_j را پیدا کرده و پهنای باند موجود بین R_j و این همسایه محاسبه می‌شود. فایل مربوط به داده‌های مورد نیاز $Task_i$ به تعداد همسایه‌های R_j و به نسبت پهنای باند آنها شکسته و اطلاعات آنها در

$$task[i].splitedData[\cdot], \dots, task[i].splitedData[k]$$

ذخیره می‌شود. با توجه به تقسیم بالا، زمان انتقال دیتای مورد نیاز task[i] به نود R_j محاسبه می‌گردد (TransferTime).

سریع‌ترین زمان ممکن برای شروع به کار task[i] براساس (TransferTime) و آخرین وضعیت نود R_j محاسبه می‌شود.

این محاسبات برای سایر R_jها بررسی می‌شود و هر نود که بهترین زمان شروع را تضمین کند، task[i] به آن نود جهت بارگیری اختصاص داده می‌شود.

فاز دو: الگوریتم ژنتیک

پس از مشخص شدن وضعیت همه‌ی وظایف ($Task_i$ها) ، لیست task[i]های بدست آمده از الگوریتم تطبیقی پویا به کروموزوم‌های متناسب با الگوریتم ژنتیک تبدیل می‌گردند.

یک جمعیت اولیه‌ی تصادفی از کروموزوم‌ها تولیدشده و سپس جواب الگوریتم تطبیقی پویا به این جمعیت اضافه می‌شود.

الگوریتم ژنتیک اجرا شده و جواب بهینه را محاسبه می‌کند.

کروموزوم بهینه به جواب مسأله‌ی زمان‌بندی برگردانده می‌شود.

شبه‌کد Pseudocode3 عملیات بهینه‌سازی مسأله زمان‌بندی پیشنهادی را با استفاده از ساختمان داده‌های تعریف شده در ابتدای این بخش بیان می‌نماید.

به‌طور کلی در مدل پیشنهادی به منظور تنظیم پارامترها، وظایف ورودی به صورت گروه‌های چندتایی در صف انتظار انباشته شده و پس از رسیدن به یک حد نصاب مشخص، زمان‌بندی می‌شوند. پارامتر حد نصاب تعداد وظایف گروهی باید طوری تنظیم شود که زمان اجرای کل در سیستم ابری به حداقل برسد (نه آنقدر زیاد که مدت زمان اجرای الگوریتم ژنتیک بالا رود و نه آنقدر کم که از قابلیت الگوریتم تطبیقی پویا بهره‌وری کامل نشود.) در فاز الگوریتم ژنتیک، اندازه جمعیت[1] را نباید خیلی مقدار زیادی قرار داد. مقدار این پارامتر رابطه غیرخطی با سرعت اجرای الگوریتم دارد، بنابراین افزایش بی‌رویه آن باعث کندشدن سرعت الگوریتم ژنتیک و در نتیجه معطل‌ماندن وظایف در حال انتظار خواهد شد. از طرفی کاهش اندازه جمعیت نیز اگرچه در افزایش سرعت پردازش اثرگذار است، اما سرعت و شانس رسیدن به جواب بهینه را کاهش می‌دهد و الگوریتم باید تکرارهای بیشتری را انجام دهد. یک پارامتر دیگر که می‌تواند اثر ویژه‌ای بر کارایی الگوریتم بگذارد، حد نصاب تکرار فایل اطلاعات است. می‌دانیم که اساس کار فاز تطبیقی پویا بازیابی داده‌ها به صورت موازی از چندین دیتاسنتر است. بدین منظور زمان‌بندی یک نسخه اطلاعات مورد نیاز برنامه‌ها را در هر یک از میزبان‌های داده‌ای کپی می‌کند. در صورتیکه اطلاعات برنامه‌ی درخواستی آنقدر کوچک باشد که با یک اتصال منفرد هم سرعت لازم و قابل قبول را برای انتقال دارد، ارسال آن به چند منبع چندان مفید نیست و بالعکس برای فایل‌های بزرگ تعداد اتصالات موازی بیشتر ضروری به نظر می‌رسد. بنابراین بهتر است این پارامتر با حجم فایل ورودی برنامه‌ها تناسب داشته باشد.

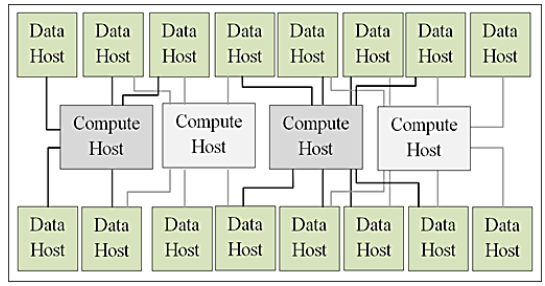
۴- یافته‌های پژوهش

محیط ابری مورد آزمایش، یک محیط شبیه‌سازی شده توسط نرم‌افزار متلب و سی‌شارپ می‌باشد. برای سادگی، وظایف ورودی بدون وابستگی در نظر گرفته شده‌اند. یعنی فرض شده است که هر وظیفه مستقل از دیگری بوده و به تنهایی اجرا می‌گردد، در واقع هر وظیفه از نظر زمان شروع به کار و داده‌های ورودی، نیازمند پایان یافتن وظیفه یا وظایف دیگر نمی‌باشد. همچنین فرض می‌شود به دلیل لحاظ‌کردن بعد مسافت در مسأله مانند شبکه‌های واقعی، هر میزبان پردازشی[2] فقط به چند میزبان داده‌ای[3] و نه لزوماً همه‌ی آنها متصل است. (شکل ۴)

1. Population Size
2. Compute Host
3. Data Host

www.SID.ir

Archive of SID



شکل ۴- وضعیت اتصال و همسایگی منابع (سیستم‌های دیتا و پردازشگر) در محیط ابری شبیه‌سازی شده مورد آزمایش

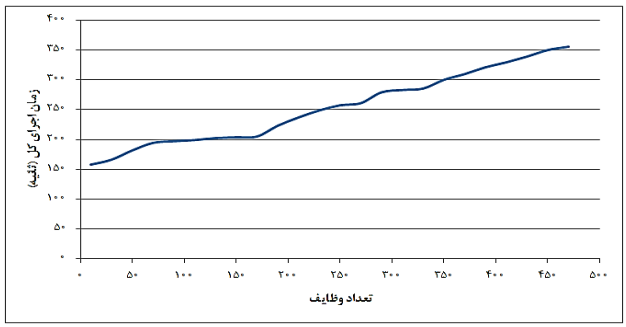
در سیستم شبیه‌سازی شده به دلیل کاهش زمان اجرای آزمایش‌ها، فقط ویژگی‌ها و مختصات اصلی هر یک از منابع ابری شامل سرعت پردازش، پهنای باند، رَم، هارددیسک در نظر گرفته شده است. اما در صورت لزوم و نیاز به تحلیل دقیق‌تر می‌توان ویژگی‌های دیگری مانند حجم حافظه ثانویه[1]، حجم و سرعت گذرگاه داده[2] و ... را نیز به سیستم شبیه‌ساز اعمال نمود از آنجا که سیستم مورد نظر یک محیط شبیه‌سازی است.

جدول ۲- مشخصات سیستم‌های پردازشی در محیط ابری شبیه‌سازی

ردیف	Bandwidth (Kbps)	RAM (GB)	CPU (GHz)	ردیف	Bandwidth ((Kbp)	RAM (GB)	CPU
C1	256	1. 75	3. 79	C11	192	1. 75	3. 60
C2	224	1	3. 25	C12	224	1	2. 51
C3	288	2	3. 26	C13	336	1	2. 61
C4	96	1	3. 10	C14	320	3	2. 90
C5	352	2. 25	2. 55	C15	160	1. 25	2. 80
C6	64	3	3. 56	C16	352	3	3. 68
C7	224	2. 75	3. 79	C17	128	1. 25	2. 89
C8	96	2	3. 29	C18	160	1. 25	2. 59
C9	256	3	3. 62	C19	192	2. 5	2. 82
C10	384	3	2. 86	C20	256	2. 5	2. 89

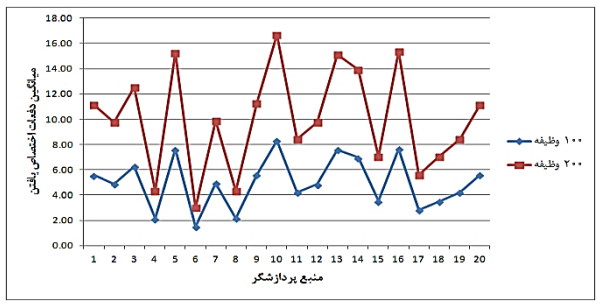
این فرض را می‌توانیم داشته باشیم که هیچ‌گونه خطا و عیبی در اتصالات داخلی شبکه بین منابع بوجود نمی‌آید و سیستم همواره برخط هستند. اولویت اول درخت kd-tree در دسته‌بندی منابع، پهنای باند در نظر گرفته شده و پس از آن رَم و سی‌پی‌یو در اولویت‌های بعدی قرار دارند. این فرض منجر به تأکید بیشتر بر مشخصه‌های قدرت سی‌پی‌یو و حجم آزاد رَم دارد، برخی منابع با چنین مشخصه‌های برتر را جایگزین می‌کند.

برای همه‌ی آزمایشات، تعداد وظایف بین ۱۰۰ تا ۵۰۰، تعداد منابع پردازش ۲۰ و تعداد منابع داده ۳۰ در نظر گرفته شده است. سرعت و قدرت سی‌پی‌یوها در بازه‌ی [۲/۳،۵/۸] گیگا هرتز و مقدار رَم‌ها در بازه‌ی [۴،۱] گیگابایت انتخاب شده است. همچنین سرعت انتقال شبکه (پهنای باند، تأخیر شبکه) در حدود [۵۰،۴۰۰] (برحسب Kbps) می‌باشد. تمامی مقادیر یاد شده با توزیع تصادفی یکنواخت تولید شده‌اند. مشخصات Taskهای ورودی (شامل حجم برنامه، مدت زمان تقریبی اجرای کامل) و داده‌های مورد نیاز آنها نیز با توزیع یکنواخت و به‌طور تصادفی ایجاد شده و برای همه‌ی آزمایش‌ها یکسانند. در این بخش، نخست نتایج مربوط اجرای دو روش تطبیقی پویا و ژنتیک ارائه می‌شود و پس از آن مشاهدات حاصل از اجرای مدل پیشنهادی آورده شده است. و در پایان به بحث و تجزیه و تحلیل درباره‌ی نتایج مدل پیشنهادی و مقایسه آنها با نتایج دو الگوریتم تطبیقی پویا و ژنتیک می‌پردازیم. قابل ذکر است به منظور مقایسه بین مدل‌های مختلف از معیار زمان اجرای کل وظایف در محیط رایانش ابری استفاده شده است. مشخصات سیستم‌های پردازشی در جدول ۱ آورده شده است.



نمودار ۱- عملکرد الگوریتم تطبیقی پویا برای اجرای تعداد مختلف

در نمودار (۲) گراف مربوط به عملکرد الگوریتم تطبیقی پویا در انتخاب منابع محاسباتی، میانگین تعداد دفعاتی که میزبان‌های پردازشی به وظایف اختصاص داده شده‌اند، نشان داده شده است. البته قابل ذکر است که به جز میانگین اختصاص یافتن منابع، می‌توان موارد دیگری را مورد بررسی قرار داد که مربوط می‌شود به برخی ویژگی‌ها و مختصات منابع که در انتخاب‌شدن آنها تأثیرگذارند.



نمودار ۲- عملکرد الگوریتم تطبیقی پویا در انتخاب منابع محاسباتی

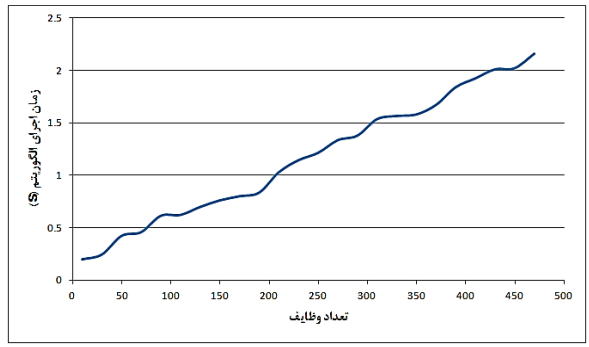
الگوریتم تطبیقی پویا با تقسیم اطلاعات مورد نیاز برنامه‌ها به نسبت پهنای باند بین منابع، آنها را از منبع داده به صورت موازی دریافت می‌کند. این ویژگی باعث می‌شود تا وابستگی بیشتری نسبت به مشخصه‌ی پهنای باند و سرعت کارت شبکه سیستم‌های ابری داشته باشد. منابع پردازشی ۵، ۱۰، ۱۳ و ۱۶ بیشترین بار به وظایف اختصاص داده شده‌اند که این نودها از سرعت شبکه (به ترتیب 352kbps، 384kbps، 336kbps و 352kbps) بهتری برخوردارند. بنابراین طبیعی به نظر می‌رسد که الگوریتم تطبیقی پویا به این خصیصه از منابع، بیشتر توجه می‌کند چرا که جهت ارسال موازی داده‌ها، به منابعی با پهنای باند بیشتر نیاز دارد. در نمودار ۳ زمان اجرای الگوریتم به ازای تعداد مختلف وظایف زمان‌بندی نشان داده شده است.

1. Cach
2. Bus

www.SID.ir

Archive of SID

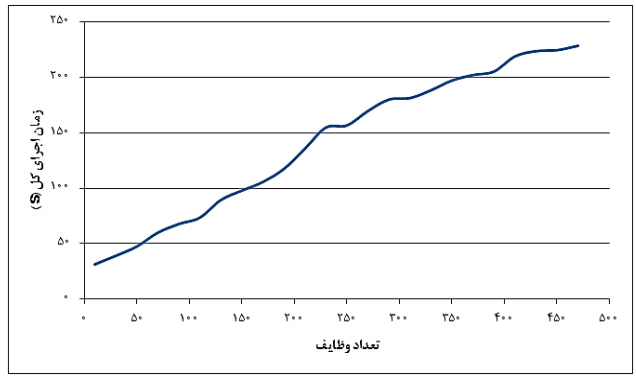
همانطور که مشاهده می‌کنید برای زمان‌بندی ۵۰۰ برنامه، ۲ ثانیه زمان صرف گردیده است. هرچند این سرعت عمل الگوریتم به تنهایی به چشم نمی‌آید و کارایی حاصل از اجرا نیز مهم است ولی با این حال در ادامه مشخص می‌شود که عامل سرعت عمل الگوریتم تطبیقی پویا در مقایسه با روش‌های ژنتیک و پیشنهادی، بسیار بهتر است.



نمودار ۳- زمان اجرای الگوریتم تطبیقی پویا به ازای تعداد مختلف وظایف.

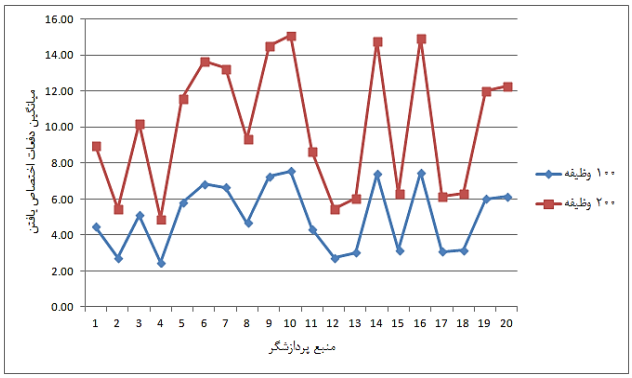
۴-۱- نتایج الگوریتم زمان‌بندی ژنتیک

مشابه بخش ۴-۱ بررسی‌ها روی سیستم ابری مورد نظر این بار توسط الگوریتم ژنتیک صورت گرفت. نمودار ۴ نشان می‌دهد با زمان‌بندی که الگوریتم ژنتیک انجام داده، حداقل و حداکثر زمان لازم برای تکمیل همه‌ی وظایف و برنامه‌های ورودی به ترتیب برابر ۳۰ و ۲۳۰ ثانیه می‌باشد. همانطور که انتظار می‌رفت این زمان‌ها نسبت به زمان‌بندی الگوریتم تطبیقی، بهتر است؛ این اختلاف هم از جستجوی دقیق‌تر الگوریتم ژنتیک نسبت به الگوریتم تطبیقی پویا، ناشی می‌شود.



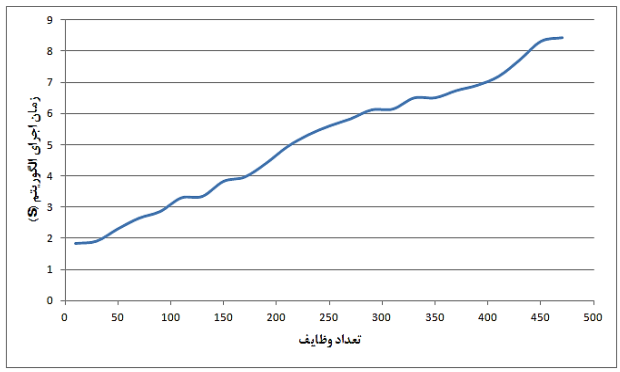
نمودار ۴- عملکرد الگوریتم ژنتیک جهت اجرای وظایف مختلف وظایف

در نمودار ۵ فراوانی انتخاب‌شدن نودهای پردازشی توسط الگوریتم ژنتیک نشان داده شده است که با نمودار مشابه آن برای الگوریتم تطبیقی پویا کمی متفاوت است. به نظر می‌رسد که روش ژنتیک منابع پردازشگر شماره ۹، ۱۰، ۱۴ و ۱۶ را انتخاب نموده است. که مشخصه توان سی‌پی‌یو هر چهار سیستم یاد شده (به ترتیب 3. 62GHz، 2. 78GHz، 2.90GHz و 3. 68GHz) است. در واقع الگوریتم ژنتیک قدرت سی‌پی‌یو را بیشتر معیار انتخاب خود قرار داده است.



نمودار ۵- عملکرد الگوریتم ژنتیک در انتخاب منابع محاسباتی) میانگین تعداد دفعاتی که میزبان‌های پردازشی به وظایف اختصاص داده شده‌اند)

اما با نگاهی به نمودار ۶ نقطه ضعف زمان‌بندی الگوریتم ژنتیک آشکار می‌شود. زمان اجرای الگوریتم ژنتیک برای یافتن یک جواب، نسبت به الگوریتم تطبیقی نسبتاً زیاد است بطوریکه حداقل ۲ و حداکثر ۹ ثانیه زمان می‌برد تا الگوریتم ژنتیک به ازای تعداد وظیفه‌های مختلف، مسأله‌ی زمان‌بندی را حل نماید. درحالی‌که روش تطبیقی پویا برای حل مسأله‌ی زمان‌بندی ۵۰۰ وظیفه‌ای در حدود ۲ ثانیه زمان صرف می‌کند، مدت زمان روش زمان‌بندی الگوریتم ژنتیک از ۲ ثانیه شروع می‌شود و در نهایت برای یک مسأله‌ی زمان‌بندی ۵۰۰ وظیفه‌ای بیش از ۸ ثانیه زمان صرف می‌کند، یعنی ۴ برابر روش تطبیقی پویا. در واقع این امر ویژگی ذاتی الگوریتم‌های تکاملی است که زمان بیشتری نسبت به روش‌های اکتشافی صرف می‌کنند.



نمودار ۶- زمان اجرای الگوریتم ژنتیک به ازای تعداد مختلف وظایف

البته اهمیت‌دادن به جنبه‌های دیگر منابع توسط الگوریتم ژنتیک، به معنای بی‌توجهی به مشخصه‌ی پهنای باند نمی‌باشد. همانطور که در نمودار ۶ مشاهده می‌شود، توزیع وظایف بین منابع نسبت به روش تطبیقی پویا کمی متعادل‌تر شده است. منطق و استدلال الگوریتم ژنتیک چنین ایجاب نموده است که در برخی موارد، از شکستن بیش از اندازه‌ی فایل‌های داده‌ها جلوگیری شود و بدین ترتیب سهم توزیع منابع با پهنای باند بالا منجر به زمان‌بندی کمتر شده است.

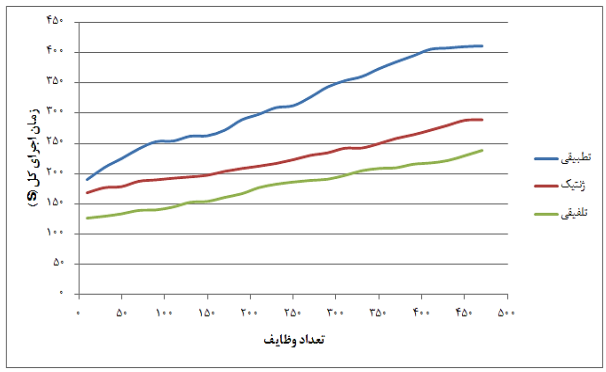
www.SID.ir

Archive of SID

۴-۲- نتایج الگوریتم پیشنهادی (تطبیقی- ژنتیک)

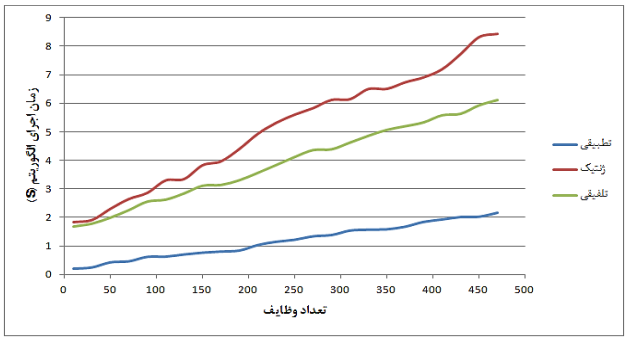
در این پژوهش، مسأله‌ی زمان‌بندی وظایف در محیط رایانش ابری مورد مطالعه قرار گرفت و ضمن بررسی مزایا و معایب دو روش مذکور، الگوریتم جدیدی با استفاده از ترکیب این دو، تحت عنوان الگوریتم زمان‌بندی تطبیقی- ژنتیک معرفی گردید. انتظار می‌رود با ترکیب دو الگوریتم ژنتیک و تطبیقی پویا، همزمان سرعت و دقت را داشته باشیم و مطابق انتظار، نتایج مشاهدات نشان می‌دهد که تقریباً به این هدف دست یافته‌ایم. همراه‌کردن الگوریتم ژنتیک با الگوریتم تطبیقی پویا، هزینه‌ی زمانی در بر داشته اما در عوض دقت جواب مسأله زمان‌بندی بهترشده است و این مسأله به معنای بهتر و سریع‌تر تکمیل‌شدن اجرای وظایف خواهد بود. از آن طرف، همراه‌کردن الگوریتم تطبیقی پویا با الگوریتم ژنتیک، به علت جواب و حدس اولیه‌ی مناسب حاصل از روش داس، سرعت رسیدن به جواب بهینه‌ی مسأله‌ی زمان‌بندی را افزایش داده است.

در نمودار ۷ مشاهده می‌شود که الگوریتم پیشنهادی از هر دو نظر سرعت و کارایی مناسب است. در واقع از نظر کارایی بهتر از الگوریتم تطبیقی پویا و از نظر سرعت بهتر از الگوریتم زمان‌بندی ژنتیک می‌باشد، به نحوی که کارایی زمان‌بندی حاصل از مدل پیشنهادی با مدل ژنتیک و مدل تطبیقی پویا مشهود است. به همان صورتی‌که قابل پیش‌بینی بود، کارایی مدل پیشنهادی بینابین دو روش دیگر قرار دارد. البته روش‌های پیشنهادی و ژنتیک کمی نزدیک‌تر به هم هستند، چرا که روش پیشنهادی در بطن خود از الگوریتم ژنتیک استفاده می‌کند البته با حدس اولیه‌ی بهتر.



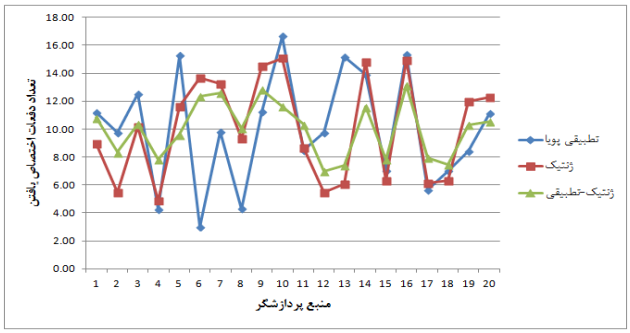
نمودار ۷- عملکرد سه روش مورد بحث از نظر مدت زمان سپری‌شده برای اجرای کل وظایف در سیستم ابری به ازای تعداد مختلفی از وظایف ورودی.

نمودار ۸ نیز به خوبی حدس و پیش‌بینی پژوهشگران را تأیید می‌کند به‌طوری‌که زمان اجرای الگوریتم پیشنهادی بهتر از الگوریتم زمان‌بندی ژنتیک بوده اما بهتر از روش تطبیقی پویا نیست. علت آن هم مشخص و بدیهی است که سرعت روش پیشنهادی و روش تطبیقی پویا تا فاز نخست مدل پیشنهادی یکسان است اما با اجرای فاز دوم، هزینه‌ی زمانی اجرای الگوریتم افزایش می‌یابد. فاز دوم مدل پیشنهادی که شامل الگوریتم ژنتیک است، به دلیل وجود جواب اولیه‌ی مناسب به‌دست آمده از فاز اول، سریع‌تر از روش زمان‌بندی ژنتیک اجرا می‌شود. همچنین نمودارها نشان می‌دهند که زمان پردازش الگوریتم زمان‌بندی تطبیقی پویا با افزایش تعداد وظایف، شیب ملایم‌تری نسبت به بقیه دارد. با وجود اینکه الگوریتم پیشنهادی پژوهش از نظر پیچیدگی زمانی، بهترین نمی‌باشد، اما از هر دو نظر سرعت و دقت به یک وضعیت میانگین رسیده است. به هر حال این موضوع نتیجه‌ی داد و ستد میان سرعت اجرا و دقت الگوریتم می‌باشد.



نمودار ۸- زمان اجرای الگوریتم‌های سه گانه جهت حل مسأله‌ی زمان‌بندی در محیط رایانش ابری

در نمودار ۹ مقایسه‌ای در ارتباط با توزیع منابع بین وظایف ابری صورت گرفته، نشان داده شده است بر مبنای آن عملکرد مدل پیشنهادی نشان می‌دهد که تقریباً به‌طور عادلانه‌تری منابع به وظایف اختصاص داده شده است. این رفتار از آنجا ناشی می‌شود که در فاز اول مدل پیشنهادی، منابع با بهترین اتصالات درون شبکه‌ای توسط درخت کی‌دی پیدا می‌شوند. الگوریتم تطبیقی پویا این منابع را به وظایف اختصاص داده و جواب حاصل به فاز دوم یعنی الگوریتم ژنتیک سپرده می‌شود. در آن مرحله، الگوریتم ژنتیک که تأکید بیشتری بر مشخصه‌های قدرت سی‌پی‌یو و حجم آزاد رَم دارد، برخی منابع با چنین مشخصه‌های برتر را جایگزین می‌کند و از آن‌جا که جواب اولیه شایستگی بالایی دارد، الگوریتم ژنتیک تمایلی به جایگزینی بی‌رویه منابع انتخاب شده در جواب اولیه ندارد.



نمودار ۹- مقایسه عملکرد سه الگوریتم زمان‌بندی بر حسب میانگین تعداد دفعات اختصاص منابع به وظایف.

مقایسه‌ی رفتار الگوریتم پیشنهادی بیان‌کننده‌ی این موضوع است که مدل زمان‌بندی پیشنهادی بر خلاف دو روش دیگر بر یک یا چند مشخصه‌ی سخت‌افزاری سیستم‌ها و منابع ابری تأکید ندارد، بلکه سعی می‌کند تا یک توازن و تعادل در روند اختصاص منابع به وظایف ایجاد

www.SID.ir

Archive of SID

نماید. این در حالیست که عملکرد الگوریتم تطبیقی پویا (نمودار۹)، نشان می‌دهد که برخی منابع بیشتر انتخاب شده‌اند به‌طوری‌که آنها منابعی هستند که پهنای باند بالاتری دارند؛ همچنین برخی منابع نیز خیلی کم اختصاص داده شده‌اند. همچنین در نمودار ۹ نشان داده شد که وضعیت الگوریتم زمان‌بندی ژنتیک کمی بهتر است اما به نظر می‌رسد بر مشخصه‌های دیگری غیر از پهنای باند تأکید دارد.

۵- نتیجه‌گیری و پیشنهادها

با ظهور سرویس‌های مبتنی بر وب و اینترنت تحول بسیار ارزشمندی در زمینه فناوری اطلاعات ایجاد شده است و با پیشرفت فناوری اطلاعات نیاز به انجام کارهای سنگین در این ارتباط به وجود آمد لذا افراد برای اینکه بتوانند کارهای سنگین اطلاعاتی و محاسباتی را بدون صرف هزینه هنگفت، از طریق خدمات انجام دهند، آخرین پاسخ فناوری به اینگونه نیازها، رایانش ابری بوده است [۲۰]. به نحوی که وجود آن به‌عنوان یک سیستم اطلاعاتی و بهبود مداوم آن منجر به افزایش کیفیت ارائه خدمت و کاهش هزینه‌های مؤثر می‌شود. در سال‌های اخیر پژوهش‌های زیادی در رابطه با رایانش ابری صورت گرفته است، بررسی مطالعات حاکی از مشکلات عمده‌ای از جمله کیفیت سرویس پایین، هزینه اجرای بالا، ضعف در تقسیم بار، ضعف در مدیریت پهنای باند شبکه ابری و عدم پردازش مناسب داده‌ها و ... بود. لذا پژوهشگران بران شدند که با توجه به نقش مؤثر و کلیدی زمان‌بندی در محیط رایانش ابری، با هدف یافتن رویکرد ترکیبی جدید از روش‌های زمان‌بندی موجود و به منظور بهینه‌کردن مسأله‌ی زمان‌بندی و کاهش زمان واکنش به درخواست‌های کاربران تحت شبکه، پژوهشی را انجام دهند. این مدل، رویکردی جدید برای بهینه‌سازی زمان‌بندی وظایف براساس ترکیبی از الگوریتم تطبیقی پویا و الگوریتم ژنتیک است به‌طوری‌که با ارائه این مدل ضمن مرتفع‌کردن معایب روش‌های مورد استفاده و داشتن دقت خوب، زمان‌بندی مناسبی برای وظایف ارائه می‌شود. همچنین در این روش برخلاف روش‌های معمول، به جای استفاده از زمان تقریبی اجرای وظایف، از نزدیک‌ترین زمان اجرای ممکن استفاده شده است. بنابراین می‌توان گفت این پژوهش تاکنون در ادبیات موضوع به روشنی مورد بحث قرار نگرفته است و در عین حال مطالعه حاضر این خلاء مطالعاتی را پشتیبانی می‌کند. به همین دلیل این پژوهش با توجه به مطالب بیان‌شده نوآوری مناسبی دارد و امید است که زمینه‌ساز پژوهش‌های آتی قرار گیرد.

به‌طور کلی در این پژوهش مسأله‌ی زمان‌بندی وظایف در محیط رایانش ابری مورد مطالعه قرار گرفت. به منظور آماده‌سازی بستر مقایسه ابتدا به بررسی دو مدل الگوریتم تطبیقی پویا و ژنتیک پرداختیم و ضمن بررسی مزایا و معایب دو روش، الگوریتم جدیدی حاصل از ترکیب الگوریتم ژنتیک و تطبیقی پویا معرفی گردید. این مدل ضمن بهره‌گیری از الگوریتم تطبیقی پویا در ابتدای پردازش، جوابی قابل قبول تولید کرد و در بخش دوم پردازش، راه‌حل تولید شده را در الگوریتم ژنتیک پالایش شد. جهت سنجش کیفیت راه‌حل‌های تولید شده براساس ویژگی‌های وظایف و منابع ابری تابع هدف (شایستگی) تعیین گردید. در نهایت الگوریتم ژنتیک موفق به یافتن جواب بهینه‌ای نسبت به سایر روش‌ها گشت، به نحوی که این مدل، با هدف پوشش‌دادن نقاط ضعف دو الگوریتم ارائه شد که به موجب آن نقطه ضعف الگوریتم زمان‌بندی تطبیقی پویا که همانا به دام‌افتادن در جواب‌های محلی است، با استفاده از الگوریتم بهینه‌سازی تکاملی پوشش داده شد، همچنین ضعف الگوریتم ژنتیک یعنی سرعت عمل پایین، با ایجاد یک راه‌حل اولیه مناسب برطرف گردید. در نتیجه الگوریتم نهایی حاصل از ترکیب این دو، جواب‌های بهتری در مدت زمان نسبتاً قابل قبول ارائه داد، به عبارتی به یک مدل زمان‌بندی کارا و مؤثر دست یافتیم.

همانطور که در بخش یافته‌های پژوهش بدان اشاره شد، آزمایش‌های مربوط به الگوریتم تطبیقی پویا، برحسب تعداد مختلفی از وظایف انجام گرفت، که با تغییر برخی از پارامترهای ورودی، تغییرات بوجود آمده در نتایج نهایی ثبت گردید. همچنین برای ثبت هر یک از مشاهدات دسته‌های مختلفی از وظایف، در نظر گرفته شد، برای هر دسته از وظایف، الگوریتم داس اجرا و زمان‌بندی شدند و برخی از منابع ابری به وظایف آن دسته اختصاص داده شد. یافته‌های پژوهش نشان داد که الگوریتم تطبیقی پویا علاوه بر سرعت عملکرد بالا، اولویت دسته‌بندی منابع تخصیص یافته را به منابعی می‌دهد که پهنای باند بیشتر و ترافیک کمتری بین آنها برقرار باشد. بدان جهت که فایل‌های اطلاعاتی مورد نیاز وظایف را با سرعت بیشتر به منابع پردازشگر برساند.

از طرف دیگر، مشابه روش تطبیقی پویا، بررسی‌ها روی سیستم ابری مورد نظر بار دیگر توسط الگوریتم ژنتیک صورت گرفت. یافته‌های پژوهش نشان داد که الگوریتم ژنتیک به منابع پردازشگر قوی‌تر، اهمیت بیشتری داده و سهم آنها را نسبت به روش تطبیقی پویا برای زمان‌بندی افزایش داده است. ولی با این حال زمان اجرای الگوریتم ژنتیک برای یافتن یک جواب، نسبت به الگوریتم تطبیقی نسبتاً زیاد بود، به‌طوری‌که برای حل مسأله‌ی زمان‌بندی ۵۰۰ وظیفه‌ای، الگوریتم تطبیقی پویا در حدود ۲ ثانیه زمان صرف می‌کند، حال آنکه زمان‌بندی الگوریتم ژنتیک از ۲ ثانیه شروع می‌شود و در نهایت برای این مسأله بیش از ۸ ثانیه زمان صرف می‌کند، یعنی ۴ برابر روش تطبیقی پویا. همچنین در این الگوریتم توزیع وظایف بین منابع نسبت به روش تطبیقی پویا کمی متعادل‌تر شده است و منطق و استدلال آن چنین ایجاب می‌کرد که در برخی موارد، از شکستن بیش از اندازه‌ی فایل‌های داده‌ها جلوگیری کند، در نتیجه سهم توزیع منابع با پهنای باند بالا منجر به زمان‌بندی کمتر شد. در عین حال هر دو روش فاکتور پهنای باند را به شکل قابل ملاحظه‌ای لحاظ کرده‌اند.

همانطور که انتظار داشتیم که با ترکیب دو الگوریتم ژنتیک و تطبیقی پویا، همزمان سرعت و دقت را داشته باشیم. مطابق انتظارمان، یافته‌های پژوهش نیز حاکی از دستیابی به این مهم بود. به‌طوری‌که استفاده از

www.SID.ir

Archive of SID

رویکرد پیشنهادی یعنی همراه‌کردن الگوریتم تطبیقی پویا با الگوریتم ژنتیک، هزینه‌ی زمانی در برداشته اما در عوض دقت جواب مسأله زمان‌بندی بهتر شده است و این مسأله به معنای بهتر و سریع‌تر تکمیل‌شدن اجرای وظایف خواهد بود. از آن طرف، به علت جواب و حدس اولیه‌ی مناسب حاصل از روش داس، سرعت رسیدن به جواب بهینه‌ی مسأله‌ی زمان‌بندی افزایش یافت. همچنین یافته‌های مربوط به میانگین تعداد دفعات انتخاب‌شدن منابع ابری نشان داد که روش پیشنهادی، رویکردی عادلانه‌تر نسبت به انتخاب منابع دارد و بر خلاف دو روش دیگر بر یک یا چند مشخصه‌ی سخت‌افزاری سیستم‌ها و منابع ابری تأکید ندارد و همچنین هزینه‌ی زمانی اجرای کل وظایف ورودی به ابر، در حالت میانگین قرار دارد؛ بدین معنا که نسبت به الگوریتم تطبیقی پویا دقیق‌تر اما با دقت کمتر، نسبت به الگوریتم ژنتیک. از طرف دیگر، سرعت ارائه یک راه‌حل زمان‌بندی مناسب، برای روش پیشنهادی نسبت به روش پویای تطبیقی کمتر و نسبت به روش ژنتیک بیشتر است.

در نهایت می‌توان گفت که استفاده از این رویکرد ترکیبی دارای یکسری مزایا و معایب است، از جمله مزایای آن علاوه بر پوشش معایب روش زمان‌بندی ژنتیک و تطبیقی پویا، می‌توان به سادگی و قابلیت فهم و درک آن، اشاره کرد به طوریکه پیاده‌سازی این مدل تلفیق شده بسیار آسان است. همچنین از آنجا که تنظیم پارامترها در هر بخش به‌طور مستقل انجام گرفت، به راحتی می‌توان نقش و تأثیر هر پارامتر از یک فاز را در روند اجرای فاز بعدی مشاهده نمود و براساس آن تصمیم‌گیری‌های لازم را انجام داد. ضمن آنکه استفاده از الگوریتم ژنتیک این قابلیت را به ما می‌دهد که اهداف گوناگونی را به تابع هدف (شایستگی) اضافه نماییم. در واقع، یک مدل چند هدفه داریم که به کمک آن هزینه‌های احتمالی که بر محیط رایانش ابری تحمیل می‌شود را در الگوریتم لحاظ نماییم و همچنین استفاده از الگوریتم تطبیقی پویا این اطمینان را می‌دهد که بتوان منابع زیادی را در محیط رایانش ابری به اشتراک گذاشت و مدیریت کرد. چرا که این کار به سادگی توسط درخت kd-tree در بطن الگوریتم تطبیقی پویا صورت می‌گیرد و هنگام یافتن راه‌حل اولیه، ساختار درختی به سرعت منابع لازم و مناسب را پیدا کرده و در اختیار زمان‌بند قرار می‌دهد. از طرف دیگر از جمله معایب آن می‌توان به این موارد اشاره کرد که اگرچه وجود الگوریتم ژنتیک در بطن الگوریتم ترکیبی اگرچه در حالت کلی یک مزیت است، اما در مواردی که محیط رایانش ابری کم بار باشد مثلاً در ساعاتی از روز که مشتریان زیادی در ترمینال‌های متصل به ابر وجود ندارند، ابعاد مسأله‌ی زمان‌بندی به اندازه‌ی کوچک است که یک روش اکتشافی مانند الگوریتم تطبیقی پویا به تنهایی می‌تواند از عهده‌ی زمان‌بندی آن برآید. بنابراین اجرای الگوریتم ژنتیک به دنبال الگوریتم تطبیقی پویا، در چنین مواردی بیهوده است. افزودن یک پارامتر کنترل بار ورودی جهت فعال یا غیر فعال کردن فاز ژنتیک از مدل پیشنهادی می‌تواند برای این موارد مشکل‌گشا باشد. همچنین درست است که الگوریتم ژنتیک در بیشتر موارد راه‌حل اولیه‌ی الگوریتم تطبیقی پویا را به سمت جواب‌های بهتر پیش می‌برد اما ممکن است در برخی موارد چنین اتفاقی رخ ندهد. البته و خوشبختانه چون این فاز از مدل پیشنهادی، مرتبط با روش‌های تکاملی است، می‌توان گزینه‌های دیگری را جایگزین الگوریتم ژنتیک نمود؛ برخی روش‌ها از جمله بهینه‌سازی اجتماع ذرات[1]، الگوریتم مورچگان[2]، الگوریتم تبریدی شبیه‌سازی شده[3] و ... ممکن است در برخی محیط‌های ابری سریع‌تر به جواب‌های بهینه دست یابند.

و در پایان با توجه به اینکه فناوری رایانش ابری خصوصاً در ایران، هنوز به بلوغ کافی نرسیده و نیازمند مطالعه و تلاش روزافزون دانش‌پژوهان کشورمان در این زمینه هستیم، پیشنهاد می‌گردد با توجه به اینکه ساختارهای درختی مانند kd-tree گزینه‌ی مناسبی برای دسته‌بندی عناصر یک مجموعه از منابع ابری هستند. استفاده از انواع دیگر درختان مانند B-tree نیز می‌تواند به‌عنوان یک روش دسته‌بندی و جستجوی منابع ابری مورد مطالعه و آزمایش و نتایج حاصل از آن مورد مقایسه قرار گیرد.

۶- مراجع

۱- محمودی، احمد و بهروزیان‌نژاد. زمان‌بندی وظایف در محیط رایانش ابری با استفاده از الگوریتم ترکیبی ژنتیک و تکاملی تفاضلی. دومین کنفرانس ملی کامپیوتر، فناوری اطلاعات و کاربردهای هوش مصنوعی، دانشگاه شهید چمران اهواز، ۱۳۹۷.

۲- درخشانیان، یلدا، میرعابدینی، جواد و هارون آبادی، علی. توازن بار در محیط رایانش ابری با در نظر گرفتن وابستگی میان وظایف و استفاده از الگوریتم ژنتیک تطبیقی. فصلنامه مهندسی مخابرات جنوب، دوره ۷، شماره ۲۴: صص ۵۸-۶۹، ۱۳۹۶.

۳- سلیمی حجت آبادی، رضا. زمان‌بندی وظایف با حفظ تعادل بار در گرید محاسباتی با استفاده از الگوریتم ژنتیک با نامغلوب و عملگرهای فازی، پایان‌نامه کارشناسی‌ارشد علوم کامپیوتر، دانشگاه غیردولتی گلبرگ، ۱۳۹۱.

۴- سپهر، فرشته، بزرگی، اشرف السادات و صدقی، شکوه. امکان‌سنجی به‌کارگیری فناوری رایانش ابری از دیدگاه کتابداران کتابخانه‌های دانشگاه‌های علوم پزشکی شهر تهران. مجله دانشکده پیراپزشکی دانشگاه علوم پزشکی تهران، شماره ۵: صص ۴۴۲-۴۲۹، ۱۳۹۵.

۵- اصغری، علی و سهرابی، محمدکریم. بررسی الگوریتم‌های زمان‌بندی منابع در پردازش ابری. دومین کنفرانس ملی و اولین کنفرانس بین‌المللی محاسبات نرَم. گیلان، ایران: صص ۱۲۸۷-۱۲۹۳، ۱۳۹۶.

۶- حبیب‌پور، نازنین و نعمت بخش، ناصر. بهبود زمان‌بندی وظایف در رایانش ابری با استفاده از الگوریتم‌های فراابتکاری. سومین کنفرانس ملی نوین در مهندسی کامپیوتر و برق، باشگاه پژوهشگران جوان و نخبگان، ۱۳۹۶.

۷- غریبی، الهام و جعفری نویمی‌پور. زمان‌بندی کارهای اولویت‌دار دارای مهلت زمانی در رایانش ابری به کمک الگوریتم بهینه‌سازی گرگ خاکستری. علوم رایانشی. شماره ۷: صص ۲۴-۳۳، ۱۳۹۶.

۸- نوری یگانه، نسیم و روستایی، رسول. زمان‌بندی وظایف در رایانش ابری. دومین کنفرانس ملی تحقیقات کاربردی در مهندسی. شیراز، ۱۳۹۶.

۹- طهماسبی، مسعود و خیام باشی، محمدرضا. ارائه مدل رایانش ابری وسایل نقلیه برای بهینه‌سازی هزینه و زمان پاسخگویی. کنفرانس بین‌المللی وب‌پژوهی. تهران، ایران، ۱۳۹۷.

1. PSO
2. ACO
3. SA

www.SID.ir

Archive of SID

۱۰- جمالی، سمیرا و شاکری، مجتبی. زمان‌بندی ایستای وظایف در محیط رایانش ابری با استفاده از الگوریتم ترکیبی بهینه‌سازی ازدحام ذرات و جستجوی. هشتمین کنفرانس هوش مصنوعی و رباتیک و دهمین سمپوزیوم بین‌المللی ربوکاپ آزاد. قزوین. ایران، ۱۳۹۷.

۱۱- میسوندی، زهرا و و جعفرزاده، نیما. ارائه یک الگوریتم جدید جهت ایجاد تعادل بار روی ماشین‌های مجازی در رایانش ابری. دومین کنفرانس بین‌المللی ترکیبات، رَمزنگاری و محاسبات. تهران، ایران، ۱۳۹۶.

۱۲- رجبی، صدیقه، رشیدا، یاسمن و مردوخی، فرهاد. مروری بر الگوریتم‌های فرااکتشافی در رایانش ابری و ارائه یک الگوریتم هیبریدی برای حل مسأله زمان‌بندی کارها در ابران. کنفرانس بین‌المللی پژوهش در علوم و تکنولوژی. تهران، ایران، ۱۳۹۵.

۱۳- رجبی، صدیقه، رشیدا، یاسمن و مردوخی، فرهاد. ارائه یک الگوریتم ترکیبی بر مبنای الگوریتم ژنتیک برای زمان‌بندی کارها در رایانش ابری. کنفرانس بین‌المللی پژوهش در علوم و تکنولوژی. تهران، ایران، ۱۳۹۵.

۱۴- حبیبی، فرشته، موتمنی، همایون و رمضانی، فرهاد.. رویکرد جدید برای حل مسأله زمان‌بندی کارها در محیط محاسبات ابری با استفاده از ترکیب الگوریتم ژنتیک و جستجوی حرام. نهمین سمپوزیوم پیشرفت‌های علوم و تکنولوژی. مشهد. ایران، ۱۳۹۳.

۱۵- رمدانی، لادن و شیروانی، میرسعید. الگوریتم زمان‌بندی وظایف مبتنی بر ژنتیک در سیستم‌های توزیعی ناهمگن. اولین کنفرانس مهندسی برق و کامپیوتر، دانشگاه آزاد اسلامی، قائم‌شهر، ایران، ۱۳۹۷.

۱۶- ابراهیمی هردورودی، حامد، نودهی، علی و ابراهیمی، حسن. ارایه روش بهبودسازی زمان‌بندی وظایف در رایانش ابری با الگوریتم گرگ خاکستری. دومین کنفرانس ملی مهندسی و فناوری ربات‌های پروازی. گرگان، ایران، ۱۳۹۷.

۱۷- اردستانی، پریا و بدیع، کامبیز. زمان‌بندی در محیط محاسبات ابری با استفاده از الگوریتم‌های تکامل. کنفرانس بین‌المللی مهندسی و علوم کاربردی. تهران، ایران، ۱۳۹۵.

۱۸- کهمره، حسین و مصلی‌نژاد، احمد. ارائه الگوریتم زمان‌بندی فرایندها در رایانش ابری مبتنی بر هوش جمعی. دومین کنفرانس بین‌المللی و سومین همایش ملی کاربرد فناوری‌های نوین در علوم مهندس. دانشگاه تربیت حیدریه، مشهد، ایران، ۱۳۹۴.

۱۹- امینی، رضا و فاطری، سهیل. روش بهبودیافته زمان‌بندی منابع در رایانش ابری با استفاده از برنامه‌نویسی ژنتیک. نهمین سمپوزیوم پیشرفت‌های علوم و تکنولوژی. مشهد. ایران، ۱۳۹۳.

۲۰- اسمعیلی رنجبر، خاطره و سلاجقه، مژده. بررسی و نقش رایانش ابری در شرکت‌های دانش‌بنیان ایران. فصلنامه رشد فناوری. شماره ۶۲. صص ۶۰- ۶۷، ۱۳۹۹.

21- Tai Wang. S ,Chuan Chen.Y and Ching Lin.Y. Energy Efficient Resource Allocation and Scheduling in Cloud Computing Platform, International Journal of Computer. Electrical, Aulomation, Control and Information Engineering, Volume 10, Issue 10, PP:1823-1827, 2016.

22- Madni, S. H. H., Abd Latiff, M. S., Abdullahi, M., Abdulhamid, S. M., & Usman, M. J. Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment. PLOS ONE, Volume 12, Issue 5, 2017.

23- Calheiros, R. N., Netto, M.A. S De Rose, C. A. F., and R. Buyya, EMUSIM. an integrated emulation and simulation environment for modeling, evaluation, and validation of performance of cloud computing applications. Software-Practice and Experience, Volume 43, Issue5, PP:595-612, 2012.

24- Ben Alla, H., Ben Alla, S., Touhafi, A., & Ezzati, A. A novel task scheduling approach based on dynamic queues and hybrid meta-heuristic algorithms for cloud computing environment. Cluster Computing. Cluster Computing., doi:10.1007/s10586-018-2811-x, 2018.

25- Gawali, M. B., & Shinde, S. K. Task scheduling and resource allocation in cloud computing using a heuristic approach. Journal of Cloud Computing, Volume 7, Issue 1, 2018.

26- Rahman, M. Adaptive Workflow Scheduling for Dynamic Grid and Cloud Computing Environment. Concurrency Computat: Pract. Exper, 25, pp: 1816–1842, 2013.

27- Abualigah, L., & Diabat, A. A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments. Cluster Computing, 2020.

28- Milan, S. T., Rajabion, L., Ranjbar, H., & Navimipoir, N. J. Nature inspired meta-heuristic algorithms for solving the load-balancing problem in cloud environments. Computers & Operations Research. , Volume110,PP:159-187, 2019.

29- Mapetu, J. P. B., Chen, Z., & Kong, L. Low-time complexity and low-cost binary particle swarm optimization algorithm for task scheduling and load balancing in cloud computing. Applied Intelligence, 2019.

30- Choudhary, A., Gupta, I., Singh, V., & Jana, P. K. A GSA based hybrid algorithm for bi-objective workflow scheduling in cloud computing. Future Generation Computer Systems, Volume 83, PP:14–26, 2018.

31- Ben Alla, H., Ben Alla, S., Touhafi, A., & Ezzati, A. A novel task scheduling approach based on dynamic queues and hybrid meta-heuristic algorithms for cloud computing environment. Cluster Computing. Cluster Computing. doi:10.1007/s10586-018-2811-x, 2018.

32- Tahrir Younis, M. & Yang, Sh. Hybrid meta- heuristic algorithms for independent job scheduling in grid computing. Applied Soft Computing, Volume 72: PP:498-517, 2018.

33- Velliangiri, S., Karthikeyan, P., Arul Xavier, V. M., & Baswaraj, D. Hybrid electro search with genetic algorithm for task scheduling in cloud computing. Ain Shams Engineering Journal:PP:1-9, 2020.

34- Saeedi,S., Khorsand,R., Ghandi Bidgoli,S., Ramezanpour,M. Improved many-objective particle swarm optimization algorithm for scientific workflow scheduling in cloud computing. Computers & industrial engineering, Volume 147: PP:159–187, 2020.

35- Gawali, M. B., & Shinde, S. K. Task scheduling and resource allocation in cloud computing using a heuristic approach. Journal of Cloud Computing, Volume 7, Issue 1, 2018.

36- Sharma,S., Tyagi,S., A Survey on Heuristic Approach for Task Scheduling in Cloud Computing. International Journal of Advanced Research in Computer Science, Volume8:PP 1089-1092, 2017.

www.SID.ir