تاریخ دریافت مقاله: ۹۶/۱۱/۲۸
تاریخ پذیرش مقاله: ۹۷/۰۶/۰۵

ارائهٔ یک مدل بدیع پیش‌بینی تطبیق‌پذیر مبتنی بر بازخورد در محیط ابری ترکیبی برای تضمین اجرای وظایف بهنگام

بابک اسماعیل پورقوچانی*
استادیار گروه مهندسی کامپیوتر– دانشگاه آزاد اسلامی واحد قوچان – قوچان – ایران
پست الکترونیکی: esmaeilpour@iauq.ac.ir

عزیزل عبدالله
استادیار دانشکده علوم کامپیوتر و فناوری اطلاعات – دانشگاه یوپی ام – سلانگور – مالزی
پست الکترونیکی: azizol@upm.edu.my

چکیده

محیط ابـری می‌توانـد به‌عنوان یک زیرسـاخت و یا تامین‌کننده سرویس برای میزبان‌های دیگر مورد استفاده قرار گیرد. یکی از مشکلات بنیادین محیط ابری غیرقطعی بودن و تاخیر تخصیص منبع در زمانی‌که سیستم تصمیم به اجرای وظایف بهنگام دارد می‌باشـد. در سیسـتم‌های بهنـگام وظایف باید در یک بازه زمانی مشـخص و دقیق اجرا و پایان یابند. در غیر این صورت سیسـتم بهنگام با شکسـت روبرو می‌شـود. در این پژوهش بـرای حل این مشـکل کلیدی در محیط ابری یک افزونه بـه همراه مدل پیش‌بینی تطبیق‌پذیر ارائه شـده اسـت که با اسـتفاده از خوشـه‌بندی وظایف و اعمال مدیریت مستقل بر روی هر خوشـه، سیسـتم را قابل پیش‌بینی می‌نماید. همچنین در این افزونه همزمان با به‌کارگیری اشـتراک سـیکل زمانی می‌خواهیم اشـتراک‌گذاری منابع بین کاربران را به‌عنوان دومین هدف دنبال نماییم. در نهایت انتظار می‌رود که این افزونه بتواند وظایف بهنگام را در مهلت زمانی مشـخص بدون هیچ خطایی به پایان برسـاند. نتایج این مقاله نشان می‌دهد افزونه بهنگام ایجاد شده توانسته به اهداف مطرح شده در این تحقیق دست یابد.

واژه‌های کلیدی: سیستم‌های بهنگام، محیط محاسبات ابری، مدیریت منابع، مدل پیش‌بینی

۱- مقدمه

سیسـتم‌های بهنگام در زمینه‌های زیادی به‌کار گرفته شده است و محیط‌های متنوعی میزبان سیستم‌های بهنگام بوده‌اند. وظایف در این سیسـتم‌ها دارای سـه مشخصه زمانـی هسـتند: زمان شـروع، مدت زمان اجـرا و زمان ضرب‌الاجل. محدودیت زمانی برای اجرای هر وظیفه یکی از نگرانی‌های سیسـتم‌های بهنگام می‌باشـد. حجم کاری بهنگام مجموعه‌ای از وظایف بهنگام می‌باشد که با یکدیگر در ارتباط بوده و در کنار یکدیگر عملیات سیستم را تکمیل می‌کنند. میزبان این وظایف بایستی بتواند تا قبل از انقضاء مهلت اجرای وظایف آن را به پایان برساند.

سـرعت مناسب در اجرای دسـتورات یکی از شروط لازم می‌باشد ولی کافی نیست. آن چیزی‌که در یک سیستم

* نویسندهٔ مسئول

بهنـگام اهمیت دارد اجرا و تحویـل وظایف پیش از زمان ضرب‌الاجل می‌باشـد. اغلب سیسـتم‌های بهنـگام با در نظر گرفتـن بدترین وضعیت ممکن منابـع مورد نیاز یک سیستم را ذخیره می‌نمایند تا بتوانند به قطعیت لازم برای اجرای وظایف بهنگام دسـت پیدا کنند. این مشکل در اغلب سیستم‌های توزیع شده که منابع به‌صورت توزیع شده و اشتراکی هستند می‌تواند مشکلات بیشتری را ایجاد نماید. محیط ابری نیز به‌عنوان نسل جدیدی از محیط‌های میزبان از این امر مستثنی نیسـت. ولیکن، محیط‌های ابری دارای مدیریـت منابع مقیاس‌پذیر هسـتند کـه به‌کاربران اجازه می‌دهد در هر زمان منابع خود را افزایش و یا کاهش دهند. در این محیط کاربران می‌تواننـد منابع مورد نیاز خود را با سـرعت جذب و منابع اضافی خـود را به محیط ابری بازگردانند. این مقیاس‌پذیـری می‌تواند این امکان را بدهد که به جای در نظر گرفتن بدترین شـرایط موجود و جذب حداکثـری منابع، منابع را بـه موقع جذب و با رفع نیاز آن را رها نماینـد. محیط‌های ابری با توجه به نوع کاربرد آن می‌توانند به‌صورت عمومی، خصوصی و یا ترکیبی باشند. در محیط‌های ابری خصوصی تخصیص منابع سـریع و بیدرنگ بوده، در حالی‌که این منابع از لحاظ کمیت محدود و گران‌قیمت می‌باشـند. در محیط‌های ابری عمومی منابع از لحاظ کمّی بیشـتر و ارزان‌تـر به‌کاربر اختصاص داده می‌شـود ولیکن مدیریت تخصیص منابع در این سیسـتم نمی‌توانـد به‌صورت بیدرنـگ این کار را انجـام دهد. در محیط‌هـای ترکیبی ابری منابع مورد نیـاز کاربران هم از منابـع اختصاصی و هم از منابـع عمومی تخصیص داده می‌شـود[۱]. ماشـین‌های مجازی می‌تواند منابع را طریق ارائه‌دهندگان منبع به‌کاربران ارائه دهد. ماشین‌های مجازی با ایجاد شفافیت می‌تواند سیاست‌ها و دستورالعمل‌ها را از سوی ارائه‌دهندگان منبع بپذیرد و آن‌ها را از دید کاربران نهایی پنهان نماید. سیاست‌های اعمال شده تامین‌کننده منابع شامل قیمت‌گذاری، زمان‌بندی و مدیریت منابع می‌باشد.

در حقیقـت تامین‌کننده منابع قوانیـن و قواعد خود را بـرای افزایش و یا کاهش مقیاس بـه محیط ابری ترکیبی تحمیل می‌کند. تامین‌کننده، منابع خود را در غالب بسته‌های پیشنهادی به زیرساخت محیط ابری ترکیبی ارائه می‌دهد. درمحیط ابری هر کاربر برای تامین منابع مورد نیاز خود به یک کارگزار نیاز دارد که آن‌ها به لیسـت منابع موجود در بسـته‌های پیشـنهادی دسترسـی داشـته و می‌تواند بـه هرکاربر این امکان را بدهد تـا بتواند از منابع موجود منابعی با قابلیت، محدودیت زمانی و بودجه انتخاب نماید. در صورتی‌کـه کاربر نتواند طرحـی را که محدودیت‌های مـورد نیازش را تامین کند انتخاب کند درخواسـت کاربر غیرقابل انجام تلقی می‌گردد و سیسـتم ابری از انجام آن صرف نظر می‌نماید[۲].

محیط ابری در درجه اول طراحی شده است که بتواند خروجی و بهره‌برداری از منابع را بیشـینه نماید، اما این محیط کاری دارای محدودیت‌هایی برای کار با برنامه‌های بهنگام دارد نظیر نداشـتن یک الگوریتم زمانبندی مناسب برای اجرای وظایف بهنگام[۳]، تأخیر در تخصیص منبع و ماشین مجازی [٤] و نبودن پیش‌بینی برای اجرای وظایف و سازوکارهای کیفیت سـرویس برای برنامه‌های بهنگام [٥]. از این رو مسـئلهٔ زیر بـرای این تحقیق در نظر گرفته شده است.

شـکل حجم کاری همـواره در حال تغییر می‌باشـد ولی مدل‌های پیش‌بینی همواره در تلاش هستند تا بتوانند یک سـاختار را در این تغییرات تشـخیص دهند. مدل‌های پیش‌بینی توانسـته‌اند به یک ابزار مناسب در محیط ابری تبدیـل گردند تا بتواننـد مدیریت مقیاس‌پذیـری منابع را به‌صـورت خودکار به‌دسـت گیرند. مدل‌هـای پیش‌بینی موجود اغلب بر اسـاس یک تک مدل طراحی شده‌اند ولی مدلی که به‌صورت تطبیق‌پذیر باشـد باید ارائه شود که بر اساس مشکل پیش رو خود را تطبیق داده و راه حل مناسب برای تشخیص الگوی حجم کاری ارائه نماید[٥].

در بخـش دوم این مقالـه به مـرور مطالعاتی که در این حوزه صورت گرفته اسـت می‌پردازیم و بخش سوم

سعی در معرفی مدل‌های پیش‌بینی که در این حوزه مورد استفاده قرار گرفته‌اند دارد. در بخش چهارم به ارائه و تشریح روش به‌کار گرفته شده در این مطالعه می‌پردازیم و سپس تنظیمات اولیه و نتایج در بخش پنجم به تصویر کشیده می‌شود. در نهایت به جمع‌بندی این مطالعه در بخش ششم می‌پردازیم.

۲-مطالب اصلی

حجم کاری وظایف بهنگام مجموعه‌ای از وظایف مرتبط با یکدیگر هستند که در صورت موفقیت در اجرا یک عملکرد را در سیستم پشتیبانی می‌کنند. میزبان منتخب باید تمامی این وظایف را در ظرف محدودیت زمانی اجرا کند [٦]. سیستم‌های بهنگام به طور معمول خود را برای بدترین شرایط آماده می‌کنند ولیکن محیطی مانند محیط ابری که امکان مقیاس‌پذیری را در مدیریت منابع به‌کاربران خود ارائه می‌نماید می‌تواند شرایط متفاوتی را برای سیستم‌های بهنگام ایجاد نماید. این به این معنی است که سیستم بهنگام می‌تواند در هنگام نیاز منابع مورد نیاز خود را جذب و منابع مازاد را رها نماید. این روشن است در صورتی‌که تمامی وظایف در ظرف محدودیت زمانی خود اجرا نگردند می‌توانند سیستم بهنگام را با شکست روبرو نمایند. در سیستم ابری به دلیل این‌که در هسته این سیستم یک زمانبندی مناسب برای اجرای وظایف بهنگام تعبیه نشده است یک نقطه ضعف برای این محیط به شمار می‌رود [۲]. بنابراین اضافه کردن یک افزونه به محیط ابری ترکیبی که بتواند این تضمین را ایجاد نماید که تمامی وظایف بهنگام در ظرف زمانی مقرر اجرا گردند یکی از اهداف این تحقیق می‌باشد. در محیط ابری مسئولیت مدیریت منابع در لایه زیرساخت قرار گرفته است. در واقع مدیریت منابع، منابع مورد نیاز هر کاربر را از طریق کارگزار آن کاربر در محیط ابری به او در غالب ماشین مجازی اختصاص می‌دهد [۷]. در لایه بُن‌سازه سرویس‌هایی که باید برای کاربر ارائه شود پایه‌ریزی می‌شود وشکل می‌گیرد. ماشین‌های مجازی می‌تواند در محیط ابری ترکیبی از محیط‌های دیگری تامین و برای ارائه سرویس به‌کاربر ارائه گردند [۸]. محیط ابری ترکیبی سیاست‌های محلی سیستم‌های توزیع شده زیرمجموعه خود را پشتیبانی و آن را از دید کاربر نهایی پنهان می‌نماید [۹]. سیاست‌های محلی برای قیمت‌گذاری، زمانبندی و مدیریت منابع مورد استفاده قرار می‌گیرند [۱۰]، سیاست‌های محلی شامل نوع و میزان منابعی می‌شود که تامین‌کننده منابع در اختیار محیط ابری ترکیبی قرار می‌دهد. همچنین، خط مشی تامین‌کننده منابع در مورد افزایش و کاهش و مقیاس خودکار منابع را شامل می‌شود که به مصرف کننده منابع تحمیل می‌شود. این سیاست‌ها اغلب در قالب مدیریت منابع و مدل اقتصادی به سیستم بالادست که همان محیط ابری ترکیبی است ارائه می‌گردد [۱۱]. در محیط ابری ترکیبی وجود تنوع در تامین‌کنندگان منابع به‌کاربر این امکان را می‌دهد که بر اساس نیاز، محدودیت زمانی وظایفی که باید اجرا شوند و بر اساس بودجه‌ای که برای اجرای آن وظایف مشخص شده است بهینه‌ترین منابع را انتخاب نماید [۲].

۲-۱- نیاز ها و خصوصیات حجم کاری بهنگام

مدیریت منابع برای اجرای وظایف متمرکز بر زمان می‌باشد و این می‌بایست با یک مدل پیش‌بینی مناسب پشتیبانی گردد. در زمان طراحی با ایجاد تغییرات در مدل پیش‌بینی بتواند وظیفه‌ای که قبل از زمان ضرب‌الاجل پایان نمی‌یابد را قبل از وقوع پیش‌بینی نماید [۱۲]. به‌علاوه حجم کاری ارسال شده به سیستم باید قابل تحلیل باشد تا بتواند نتایج تصمیمات زمان‌بندی و تغییرات را پیش‌بینی نموده و به سطح قابل قبولی از اجرا دست یابد [۱۳]. یکی از مسائلی که باید در سیستم‌های بهنگام حل شود مشکل عدم قطعیت سیستم است که با به‌کارگیری مدل پیش‌بینی و افزودن یک زمان لغزش به وظایف می‌توان آن را حل نمود[۱۵-۱٤]. در نتیجه سیستم‌های بهنگام باید بتوانند احتمالاتی را که در زمان اجرای وظایف رخ می‌دهد پیش‌بینی و تضمین نماید که تمام زمان‌های مهم مد نظر قرار می‌گیرد. باروس و

سایرین در تحقیقات خودشان به این نتیجه دست یافتند که دستاوردهای گذشته را نمی‌توان در محیط جدید استفاده نمود به دلیل آن‌که آن‌ها به دانش گذشته اجرای حجم کاری گذشته نیاز دارند و این به این معنی است که میزبان سیستم باید قطعی باشد. بنابراین محیط‌های غیرقطعی نظیر محیط ابری اساسا دارای مشکل می‌باشند. مفهوم قطعیت به این معناست که رفتار سیستم باید قابل پیش‌بینی از جنس مقدار و زمان باشد. رفتار آتی سیستم معلول شرایط فعلی سیستم می‌باشد. بنابراین دست‌یافت‌هایی نظیر داشتن منابع ذخیره بر اساس قراردادهای از پیش تعیین شده با تعیین موارد استفاده و یا گرفتن بازخورد از شرایط کنونی سیستم به‌عنوان راه حل برای سیستم‌های بهنگام ارائه می‌شود [٦].

اما شرایط در سیستم‌های غیر قطعی به این شکل می‌باشد که مشتری هیچ وظیفه‌ای ندارد که زمانبندی وظایف خود را برای سیستم از قبل ارسال نماید و یا نمی‌داند چه حجم از منابع را در چه زمانی نیاز دارد. بنابراین باید روشی را به‌کار برد که رفتارهای قطعی در این سیستم ایجاد تا برای اجرای وظایف بهنگام مناسب گردد [١٦]. همچنین، در خصوص طراحی سیستم‌های بهنگام قبل از طراحی زمانبندی برای آن باید بررسی شود که سیستم چگونه می‌تواند در مورد عدم قطعیت مقاومت کند. با تخمین زدن احتمال وقوع رفتارهای غیر قطعی می‌توان مقاومت سیستم در زمان بروز اتفاق های ناخواسته را افزایش داد. هر چند به دلایل زیر ممکن است سیستم نتواند در زدن این تخمین به خوبی عمل نماید [١٧،١٣].

- پایه محاسباتی به درستی تعریف نشده است
- در صورت بروز خطای سخت‌افزاری و شکست سخت افزاری سیستم دچار اختلال یا شکست گردد.
- مفهوم زمان و محدودیت زمانی به درستی تعریف نشده باشد.
- سیستم دارای ساختاری غیر قطعی در طراحی باشد.

سیستم نیاز دارد عدم اطمینان را در سطح خصوصیت‌های کمّی برای مدل پیش‌بینی در سیستم‌های بهنگام تعریف نماید. در ادامه، یک مدل جامع باید ایجاد گردد تا هر دو خصوصیت ١- عدم اطمینان در دامنه‌های دلخواه از داده‌ها برای پیش‌بینی طولانی مدت و ٢- عدم اطمنیان موقت مربوط به پردازش‌های برنامه‌ریزی و پیش‌بینی اتفاقات که ممکن است در یک سیستم غیر قطعی در یک دوره زمانی رخ دهد شامل شود [١٩-١٨]. برای سیستم‌های بهنگام بزرگ این می‌تواند خیلی جالب باشد که یک الگوریتم سطح بالا برای حل همه مشکلات تعریف شود که هم مفاهیم بهنگام را در ذیل خود داشته باشد و هم بتواند بیشترین سطح به‌کارگیری منابع را داشته باشد[١٣].

٢-٢ شناسایی الگوی حجم کاری

کارآیی یک مدل پیش‌بینی بیشترین وابستگی را به حجم کاری دارد[٢٠] و بر اساس داده‌های گذشته کارآیی فعلی سیستم را می‌توان شکل داد. این معیارها می‌تواند میزان به‌کارگیری پردازنده، زمان انتظار، هزینه ماشین مجازی و زمان پاسخ باشد [٢١]. تاثیر تغییرات می‌تواند به‌صورت دقیق با استفاده از حجم کاری گذشته شناسایی شده تا ریسک تفاوت اجرای وظایف را به حداقل برساند.

برای این منظور خصوصیت حجم کاری باید به خوبی تعریف شود و اگر این تعریف به خوبی انجام شود در نتیجه تامین‌کننده می‌تواند به راحتی حجم کاری را مدل نماید [٢٢]. در مقالات مطالعه شده سه تکنیک کلی برای تقریب وظیفه آتی ارائه شده است: ١- ایجاد پروفایل برای حجم کاری [٢٣] ٢- مدل‌سازی حجم کاری[٢٣] ٣- پیش‌بینی حجم کاری. تکنیک‌های آمارگیری یکی از مطمئن‌ترین راه‌ها برای استخراج اطلاعات درون حجم کاری می‌باشد[٢٦-٢٣]. آنچه که مشخص است با روش دوم یعنی همان مدل‌سازی می‌توان احتمال را با دقت بهتری پیش‌بینی کرد ولی امکان مدل‌سازی برای تمامی برنامه‌ها و یا حجم‌های کاری وجود ندارد[٥-٤، ٣٠-٢٧].

جدول ۱: نمونه‌ای از مدل‌های زمانی بازخورد دار

مدل	توضیحات مربوط به مدل
MA(q)	$\{y(t) = B_{1*q}\epsilon(t)\}$
MEAN(q)	$\{y(t) = \mu + B_{1*q}\epsilon(t)\}$
ARMA(p,q)	$\{y(t) = A_{1*p}x(t) + B_{1*q}\epsilon(t)\}$
ARIMA(p,q,d)	$AR(p)\left((1-L)^d y(t)\right) = MA(q)$

۲-۳ ارزیابی در مدل پیش‌بینی

ارزیابی داده‌ها برای محاسبه خطاهای مدل ارائه شده مورد استفاده قرار می‌گیرد. معیارهای زیادی برای ارزیابی وجود دارد که مورد استفاده قرار می‌گیرد که می‌توان به‌عنوان مثال به خطای میانگین، میانه، بیشترین، کمترین و میانگین مطلق اشاره کرد. این معیارها کمک می‌کنند که اختلاف پیش‌بینی صورت گرفته با واقعیت را محاسبه کرد. اغلب مدل‌های پیش‌بینی دارای نقاط وارسی برای ارزیابی خودشان در هنگام اجرا هستند و در صورتی‌که خطای پیش‌بینی زیاد باشد ضرایب پیش‌بینی برای مرحله بعدی مناسب‌سازی می‌شوند. در واقع ارزیابی از پیش‌بینی می‌تواند به پایداری مدل پیش‌بینی کمک نماید [۳۱].

۳- کارهای مرتبط

اغلب کارهای مرتبط معمولا دارای روش‌های مشابه برای یک مسئلهٔ خاص هستند. در این بخش از تحقیق به بررسی طراحی آن‌ها و میزان موثر بودن طراحی آن‌ها به‌عنوان راه حل یک مسئله می‌پردازیم. این مدل‌ها بر اساس تنظیماتی که پژوهشگران انجام داده‌اند پیاده‌سازی و مقایسه شده‌اند.

۱-۳ مدل‌های پیش‌بینی بازخورد دار

بازخورد دارای مزایایی برای کارآیی سیستم است و این به سیستم کمک می‌کند که خروجی آتی خود را با استفاده از بازخورد جاری بهبود دهد. سیستم کنترلی بازخورد چندین نقطه ارزیابی را ایجاد می‌نماید تا مقدار پیش‌بینی را به واقعیت نزدیک نماید. این نقاط ضرایب ارزیابی را بهبود می‌دهند [۳۲]. سیستم کنترلی بازخورد دارای تعاملاتی با محیط خود در دو جهت می‌باشد. عملی که توسط محرک تولید می‌شود به طور مستقیم در ارتباط با اطلاعات جاری است که هم اکنون حس شده است.

مدل‌های بازخورد دار موجود در جدول یک، مدل‌های زمانی هستند که از خطای گذشته برای تولید داده‌های آتی استفاده می‌کنند. در [۳۰] از مدل حرکت معدل در محیط ابری برای پیش‌بینی میزان درخواست ماشین مجازی کاربران و از مدل میانه در تحقیق [۳۳] برای پیش‌بینی دینامیکی و برخط میزان منابع مورد نیاز در محیط ابری استفاده شده است و مدل رگرسیون خودکار حرکت معدل در تحقیق [٤] به‌عنوان تشخیص دهنده رفتارهای غیر معمول و در تحقیق [۳٤] برای تخمین میزان درخواست کاربران در یک سیستم بهنگام IPTV مورد استفاده قرار گرفته است. مدل ARIMA نیز در یک تحقیق برای تشخیص رفتارهای غیر معمول صفحات وب مورد استفاده قرار گرفته است [۳۱].

مدل SVMGA

طبیعت مدل‌های پیش‌بینی مبتنی بر SVM اینست که برای کار با سیستم‌های غیر خطی چند عاملی مورد استفاده قرار می‌گیرند و این می‌تواند به ایجاد ارتباط غیر خطی برای تغییرات عامل‌ها کمک نماید. در نتیجه مدل می‌تواند تغییرات را با دقت بالایی به کمک تغییرات اجزا پیش‌بینی نماید. تعیین پارامترهای SVM بسیار دشوار است بنابراین مدل ارائه شده در تحقیق[۳۵] نشان داده است که پارامترهای SVM دارای تاثیر مستقیم بر روی دقت SVM هستند. در این مدل در صورتی‌که ارزیابی نشان دهنده انحراف نتایج از واقعیت باشد، با استفاده از الگوریتم ژنتیک پارامترهای مورد استفاده برای توزیع داده‌ها را اصلاح می‌نماید.

۴- روش تحقیق

در این تحقیق سعی بر این است که با ارائه مدل پیشنهادی بتوان اجرای وظایف بهنگام را در سیستم محاسبات ابری ترکیبی در مقایسه با سایر مدل‌های ارائه شده کارآمدتر نمود. در بخش قبل در مورد سایر مدل‌های

پیش‌بینی که در محیط ابری توسط سایر محققان برای مدیریت منابع و پیش‌بینی میزان تقاضاهای کاربران به‌کار گرفته شده‌اند پرداخته‌ایم. در نتیجه با توجه به نقاط ضعف و قوت آن‌ها سعی شده است با انجام بهینه‌سازی، میزان موفقیت آن‌ها در اجرای وظایف بهنگام در محیط ابری افزایش پیدا کند. در بررسی‌های انجام شده در بخش مرور ادبیات مشخص شد که سیستم بهنگام نیازمند یک واکنش بهنگام و تطبیق‌پذیر با توجه به شرایط کنونی سیستم می‌باشد.

در این تحقیق ابری ترکیبی به‌عنوان میزبان در نظر گرفته شده است. در نتیجه این اقدام، کاربر این امکان را دارد که منابع مورد نیاز خود را هم از محیط ابری خصوصی و هم از محیط ابری عمومی تامین نماید. لایه زیرساخت وظیفه اجرای مدیریت منابع را در محیط ابری بر عهده دارد. در این لایه منابع فیزیکی به هسته این زیرساخت تحویل و از طریق ماشین‌های مجازی در اختیار کاربران قرار می‌گیرند. محیط ابری ترکیبی همان‌طور که قبلا گفته شد می‌تواند سیاست‌های محلی تامین‌کنندگان محلی خود را تامین نماید ولی آن‌ها را از دید کاربران نهایی خود پنهان نگاه می‌دارد. تامین‌کنندگان محلی منابع خود را در غالب طرح‌هایی به محیط ابری ترکیبی ارائه می‌نمایند.

پیاده‌سازی محیط ابری ترکیبی اولین قدم در پیاده‌سازی در محیط شبیه‌سازی کلودسیم[1] می‌باشد و برای این منظور کارهای زیر در این محیط پیگیری شده است ؛ ۱- طراحی چندین مرکز داده‌ها با منابع کمتر و محدودیت اعطای منابع به‌عنوان مرکز داده‌های خصوصی ۲- طراحی چندین مرکز داده‌ها با منابع بیشتر و عدم محدودیت دسترسی برای کاربران به‌عنوان مرکز داده‌های خصوصی ۳- طراحی میزبان درون مرکز داده و تعریف زیرساخت سخت‌افزاری آن ٤- تعریف ماشین مجازی و اعطای میزان مشخصی از منابع به آن ٥- برنامه‌ریزی و طراحی سرویس‌ها و اعطای وظایف حذف، ایجاد، انتقال و مهاجرت ماشین‌های مجازی به آن.

1- cloudSim

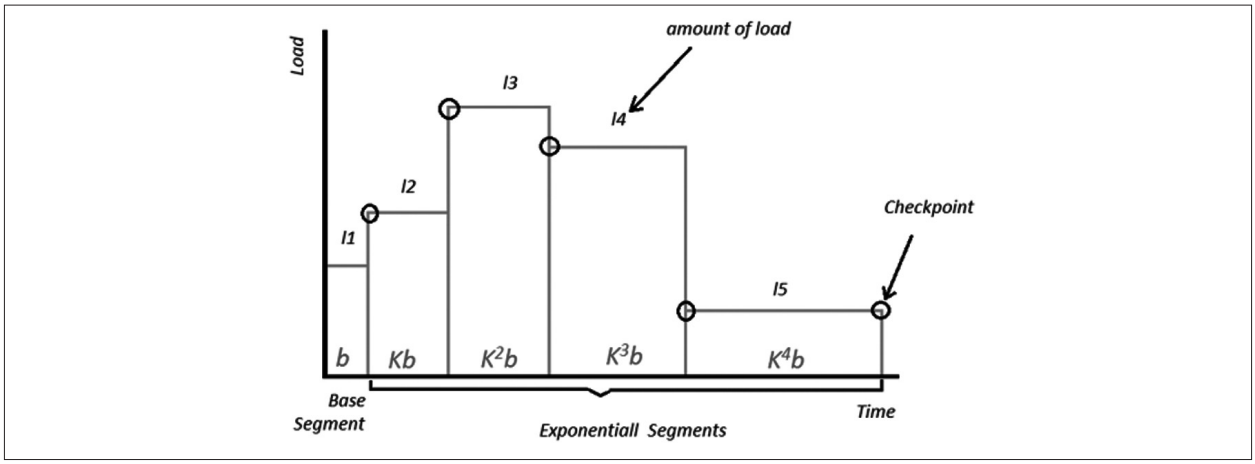
در قدم بعدی در محیط شبیه‌سازی اقدام به طراحی بخشی برای دریافت حجم کاری وظایف بهنگام شده است. در این بخش وظایف بهنگام به طور کامل پایش شده و پارامترهای کلیدی آن استخراج می‌گردد. داده‌های به دست آمده نرمال‌سازی، فیلتر و تسویه شده است. در نهایت داده‌های مورد نیاز برای مدل ارائه شده فراهم می‌نماید. هر وظیفه بهنگام سه خصوصیت زمانی مهم را دارا می‌باشد. به‌عنوان مثال وظیفه T_i دارای زمان ورود ta، زمان مورد نیاز برای اجرا t_x و زمان ضرب‌الاجل $t_{deadline}$ می‌باشد. سیستم در صورتی موفق عمل می‌کند که زمان پایان اجرای دستورالعمل قبل از زمان ضرب‌الاجل باشد. در نتیجه رابطه زیر می‌بایست برای همه وظایف صادق بوده تا در پایان سیستم بهنگام بدون شکست اجرای وظایف را به پایان برساند.

$$t_{deadline} \geq t_a + t_x \quad (1)$$

در این تحقیق از زیرساخت ابری ترکیبی برای پیاده‌سازی سیستم بهنگام استفاده شده است. زیرساخت سیستم ابری ترکیبی در محیط شبیه‌سازی کلودسیم و با زبان برنامه‌نویسی جاوا پیاده‌سازی شده است.

یکی از مواردی که در این پیاده‌سازی مدنظر قرار گرفته است مسئلهٔ مهم خوشه‌بندی است که در تحلیل‌های ایستا به‌کار می‌رود، به دلیل آن‌که می‌تواند سرباری را که بر روی پایش است کاهش دهد. در این تحقیق از دو پارامتر کلیدی اندازه وظیفه[2] و تعداد پردازنده مورد نیاز استفاده شده است و داده‌ها به‌صورت دو بعدی خوشه‌بندی می‌گردند. روش خوشه‌بندی K-Mean برای انجام این کار در این سیستم انتخاب شده است. در واقع روش خوشه‌بندی K-Mean برای خوشه‌بندی داده‌ها و اشیاء مورد استفاده قرار می‌گیرد [۲۸،۲۰]. در این روش خوشه‌بندی وظایف بر اساس شباهت‌هایی که در این دو پارامتر هر وظیفه وجود دارد در یکی از K خوشه قرار می‌گیرند. الگوریتم خوشه‌بندی K-Mean در مجموعه الگوریتم‌های با پیچیدگی زمانی NP-Hard به شمار می‌رود. پارامترهایی نظیر ابعاد،

2- Task size



شکل ۱: قطعه‌بندی حجم ورودی وظایف و قرار گرفتن نقطه وارسی در انتهای آن

تعداد نمونه‌ها و تعداد خوشـــه‌ها در میـــزان پیچیدگی این روش تاثیرگذار است [۲۸،۲۰].

$$l_c' = K - Mean(c) \quad (۲)$$

در رابطه بالا وظایف در خوشه‌های متفاوت تقسیم شده‌اند و این باعث شـــده است که هر خوشـــه c پس از پایان کار K-Mean میزان بار مشخصی به اندازه l_c داشته باشد.

در ســـومین مرحله همان‌طور که در شـــکل ۱ نمایش داده شـــده حجم کاری در خوشـــه به‌صـــورت قطعاتی با اندازه مشخص تقسیم می‌شود. در پایان هر قطعه یک نقطه وارســـی[۳] قرار گرفته اســـت. اولین قطعه را در هر خوشه در اصلاح قطعه زیربنایی می‌نامیم. ســـایر قطعات از لحاظ اندازه بزرگ‌تر و نسبتی نمایی نسبت به قطعه قبل از خود دارد. بنابراین، قطعه دوم به بعد را قطعات نمایی می‌نامیم. انـــدازه هر قطعه نمایی از قطعه ما قبل خود m بار بزرگ‌تر است.

در رابطـــه زیر n تعداد قطعـــات نمایی و b اندازه قطعه زیربنایـــی می‌شـــود بنابراین اندازه قطعـــه i را می‌توان از رابطه زیر محاسبه کرد.

$$SE_i = K^i b \,, i \in \{1,2,..,n\} \quad (۳)$$

در نهایـــت حجم کاری وارد شـــده به قطعات را می‌توان از رابطه سری هندسی زیر محاسبه کرد.

$$b + bk + bk^2 + \cdots + bk^n = T \quad (۴)$$

3- Checkpoint

$$T = \frac{(1 - k^{n+1} b)}{(1 - k)} \quad (۵)$$

از رابطـــه میانگیـــن متحـــرک می‌تـــوان در یـــک مدل بازخورددار اســـتفاده کرد و این نشان می‌دهد که این مدل چگونـــه می‌تواند میزان بار وارد شـــده را در بازه زمانی i پیش‌بینی نماید.

$$l_{i,c} = l_{1,c} + \frac{t_{i-1} - t_0}{t_1 - t_0} l_{1,c} \quad (۶)$$

$$\begin{aligned} l_{i,c} &= l_{1,c} + \frac{t_{i-1} - t_0}{t_1 - t_0} l_{1,c} \\ &= l_{1,c} + \frac{\frac{(1 - k^i) b}{1 - k} - b}{kb - b} l_{1,c} \\ &= l_{1,c} + \frac{(k^{i-1} - 1) k}{(k - 1)^2} l_{1,c} \end{aligned} \quad (۷)$$

رابطه فوق برای تمامی خوشـــه‌ها در انتهای هر خوشـــه اجرا می‌شـــود تا بتواند میزان بار را در قطعه بعدی تخمین بزند. در انتهای اولین قطعه از K-Mean برای خوشـــه‌بندی و توزیع وظایف در K خوشه استفاده شده است تنها به این دلیل که در انتهای اولین خوشـــه تعداد نمونه‌های ورودی محدود اســـت و این کار باعث می‌گردد که پیچیدگی زمانی روش K-Mean کاهـــش یابد در مراحل بعـــدی وظایف با استفاده از رابطه پیکتورین (رابطه (۸)) کوتاه‌ترین فاصله بیـــن پارامترهای وظیفـــه وارد شـــده و پارامترهای مرکز خوشه را محاســـبه کرده تا نزدیک‌ترین خوشه به خود را

به‌دست آورد. در نهایت پس از به‌دست آمدن کوتاه‌ترین فاصله وظیفه به آن خوشه منتقل می‌گردد. پارامترهای مرکزی خوشه c شامل میانگین تعداد پردازنده‌های مورد نیاز ($M_{pe,c}$) و میانگین اندازه‌های وظایف درون خوشه ($M_{MI,c}$) می‌گردد و پارامترهای هر وظیفه وارد شده به سیستم شامل تعداد پردازنده‌های مورد نیاز هر وظیفه (t_{pe}) و اندازه هر وظیفه (t_{MI}) می‌باشد.

$$d^c = \sqrt{(M_{pe,c} - t_{pe})^2 + (M_{MI,c} - t_{MI})^2} \qquad (۸)$$

پس از مشخص شدن خوشه هدف و اضافه شدن وظیفه ورودی به آن با استفاده از رابطه میانگین پیوسته می‌توان مقادیر میانگین تعداد پردازنده‌های مورد نیاز ($M_{pe,c}$) و میانگین اندازه‌های وظایف درون خوشه ($M_{MI,c}$) را بروزرسانی کرد.

در نهایت در آخرین قطعه و در نقطه وارسی آخر پارامترهای کارآیی سیستم محاسبه می‌شود تا سیستم بتواند تخمینی از میزان منابع مورد نیاز خود داشته باشد. در نهایت نتیجه این مدل پیش‌بینی توسط مدل اقتصادی برای جذب منابع مورد نیاز به‌کار گرفته خواهد شد.

۵- روش‌های جمع‌آوری داده

این تحقیق از مجموعه داده‌های گوگل کلاستر که در تاریخ ۲۰ می ۲۰۱۲ ذخیره شده استفاده کرده است.

۵-۱ منبع داده

منبع داده‌های مورد استفاده در این تحقیق توسط گوگل در سال ۲۰۱۵ به‌صورت عمومی از اندازه‌گیری‌هایی که از محیط ابری خود انجام داده منتشر شده است. در این منبع داده داده‌ها از لحاظ اهمیت به ۱۰ رده با اولویت‌های مختلف دسته‌بندی شده‌اند [۳۶]. وظیفه‌هایی که اولویت بالاتری دارند می‌توانند منابع را آسان‌تر از وظایفی که اولویت پایین‌تری دارند جذب نمایند. این منبع داده حاوی ۵۰ میلیون وظیفه است ولیکن همه این وظایف به دلایل زیر قابل استفاده نیستند. ۱- به‌صورت مناسب گردآوری و ذخیره نشده‌اند ۲- بخشی از اطلاعات آن‌ها از دست رفته است ۳- به‌صورت غیر قابل باوری دارای زمان اجرای بالایی می‌باشند ٤- محدودیت‌های ابزار شبیه‌سازی را نقض می‌کنند ٥- در زمان اجرا دچار مشکل شده‌اند و نتوانسته‌اند فرآیند اجرای خود را تکمیل نمایند.

۵-۲ داده‌های خوشه‌ای گوگل

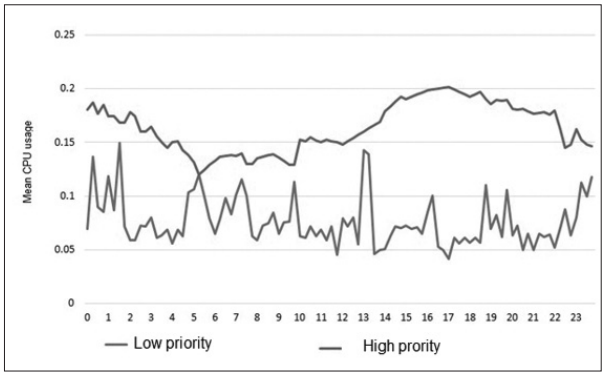
تمامی اطلاعاتی که در این مقاله مورد استفاده قرار گرفته در ماه می ۲۰۱۲ ذخیره آن به پایان رسیده است. این حجم کاری نشان می‌دهد چند منبع توسط هر وظیفه مود تقاضا قرار گرفته و چند منبع دریافت شده است و به‌علاوه این اطلاعات می‌تواند مشخص کند هر ده رده از اولویت‌بندی‌ها چه حجمی از اطلاعات را دریافت کرده‌اند. هر وظیفه برای اولویت‌بندی یک عد صحیح دارد که مقداری بین ۰ الی ۹ را می‌تواند به خود بگیرد که مقدار صفر نشان دهنده پایین‌ترین اولویت و مقدار ۹ نشان دهنده بالاترین اولویت می‌باشد. در این تحقیق [۳٦] وظایف به سه دسته اولویت بالا، متوسط و پایین تقسیم‌بندی شده‌اند. شکل ۲ نشان دهنده دو اولویت بالا و پایین در یک روز خاص می‌باشد و تفاوت میانگین مصرف پردازنده در بین این دو اولویت را به نمایش گذاشته است. پنجره جمع‌آوری این اطلاعات برای گرفتن میانگین ۳۰۰ ثانیه (۵ دقیقه) بوده است.

۵-۳ متغیرهای اندازه‌گیری عملکرد

یکی از مواردی که باعث انتخاب یک الگوریتم از بین چندین الگوریتم ارائه شده می‌شود اینست که بدانیم کدام خصوصیت یک الگوریتم دارای اهمیت است که ملاک انتخاب قرار گیرد. بر اساس تحقیقاتی که صورت گرفته، متغیرهای اندازه‌گیری عملکرد زیر مشخص شده و معیار انتخاب ما در این تحقیق قرار گرفته‌اند و بر اساس آن مقایسه با سایر الگوریتم‌های دیگر را انجام می‌دهیم.

دقت

برای اندازه‌گیری دقت از معیار اندازه‌گیری عملکرد R^2



شکل ۲: متوسط مصرف پردازنده در دو اولویت بالا و پایین با پنجره ۳۰۰ ثانیه در طول ۲۴ ساعت

استفاده شده است که می‌تواند دقت پیش‌بینی را به کمک آن اندازه‌گیری و میزان منطبق بودن مقدار پیش‌بینی با مقدار اصلی را مقایسه کرد. برای اندازه‌گیری این معیار از رابطه ریاضی زیر استفاده می‌کنیم.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y} - y_i)^2}{\sum_{i=1}^{n}(\hat{y} - \bar{y})^2} \qquad (۹)$$

مقدار $\bar{y}$ میانگین و مقدار y میانه داده‌ها می‌باشد که در رابطه فوق محاسبه و مقدارگذاری می‌شوند. همچنین مقدار y_i مقدار خروجی مدل پیش‌بینی می‌باشد و مقدار n هم تعداد مشاهدات می‌باشد. مقدار R^2 همواره در بازه $[۰,۱]$ قرار می‌گیرد و این مدل بر روی رگرسیون‌های خطی اعمال می‌گردد. در حقیقت R^2 دقت پیش‌بینی است که می‌تواند نشان دهد که نتایج پیش‌بینی مدل به چه میزان بر روی داده‌های واقعی قرار گرفته است.

بهره وری[4]

خطای مجذور متوسط ریشه[5] اغلب برای اندازه‌گیری تفاوت بین مقدار پیش‌بینی شده و مقدار مشاهده شده از محیط است، این تفاوت‌ها در واقع مجذور باقی مانده‌ها هستند که به‌صورت زیر محاسبه می‌شود. هر چقدر مقدار خطای مجذور متوسط ریشه کوچک‌تر باشد مدل بهره‌وری بالاتری را برای مدل پیش‌بینی نشان می‌دهد.

$$RMSE = \sqrt{\frac{(l_{observed} - l_{predicted})^2}{n}} \qquad (۱۰)$$

4- Efficiency
5- Root Mean Square Error (RMSE)

متوسط درصد خطای نسبی

متوسط درصد خطای نسبی که متوسط درصد انحراف مطلق نیز شناخته می‌شود[6] می‌تواند دقت مدل پیش‌بینی را بر اساس روش اندازه‌گیری آماری در چهارچوب زمانی t اندازه‌گیری نماید.

$$\bar{\epsilon_t} = \sum_{j=1}^{n}\left|\frac{\hat{y} - y_j}{y_j}\right| \qquad (۱۱)$$

مقدار $\bar{y}$ میانگین و مقدار $\hat{y}$ مقدار میانه داده‌ها می‌باشد که در رابطه فوق محاسبه و مقدارگذاری می‌شوند. همچنین مقدار y_j مقدار خروجی مدل پیش‌بینی می‌باشد و مقدار n هم تعداد مشاهدات می‌باشد.

بهره‌برداری[7]

برای محاسبه میزان بهره‌برداری، در این تحقیق از ابزار محاسبه بهره برداری شبیه‌ساز کلودسیم استفاده شده است. این ابزار نشان می‌دهد که به چه میزان منابع توسط وظایف به‌کار گرفته شده است. در شبیه ساز کلودسیم تعداد قابل توجهی سیاست واگذاری ماشین مجازی وجود دارد که این سیاست‌ها مشخص می‌کند که چگونه کنترل‌کننده باید ماشین‌های مجازی را بین میزبان‌های موجود توزیع نماید. هر یک از این سیاست‌های واگذاری ماشین مجازی می‌تواند تاثیر زیادی بر روی بهره وری و یا مصرف انرژی داشته باشد. در این تحقیق از سیاست آستانه دوایستایی[8] استفاده شده است. در این سیاست دو سطح آستانه تعریف شده است تا نشان دهد که میزبان دچار اضافه و یا کاهش بار شده است یا خیر. زمانی‌که میزان بهره‌برداری صد درصد باشد و صف وظایف پر باشد سیستم دچار اضافه بار می‌شود و در صورتی‌که میزان بهره‌برداری از یک مقدار کمتر باشد سیستم با کاهش بار رو برو می‌شود. میزان بهره‌برداری بر اساس رابطه (۱۱) محاسبه می‌شود. این رابطه در واقع نسبتی است بین مدت زمانی‌که پردازنده دارای بار نمی‌باشد به

6- Mean Absolute Percentage Deviation(MAPD)
7- Utilization
8- Double Static Threshold (DST)

مدت زمانی‌که پردازنده دارای مقداری بار برای کار کردن می‌باشد.

$$\text{Utilization} = 1 - \frac{\text{The period CPU has not any load}}{\text{The period CPU has some load}} \qquad (۱۱)$$

شکست

در یک سیستم، شکست زمانی روی می‌دهد که سیستم از اجرای وظایف عادی خود فاصله می‌گیرد. میزان وظایفی را کــه امکان به پایان رســاندن اجرای خــود قبل از زمان ضرب‌الاجل ندارند، میزان شکست می‌نامند. بر اساس نوع سیســتم بهنگام که می‌تواند سخت، متوسط و یا نرم باشد سیستم می‌تواند توانایی تحمل شکست را داشته باشد و یا در برخورد به آن سخت بوده ودچار شکست کامل شود. بنابرایــن، این متغیر اندازه‌گیری عملکرد یکی از معیارهای اندازه‌گیری مهم برای تشــخیص موفقیت مدل ارائه شــده برای یک سیســتم بهنگام می‌باشد که بر اساس رابطه زیر محاسبه می‌شود[٦].

$$\text{Failure} = \begin{cases} 1 & \forall i \; \exists \; t - \text{execution}_i < t - \text{deadline}_i \\ 0 & \text{else} \end{cases} \qquad (۱۲)$$

تنظیمات محیط کاری

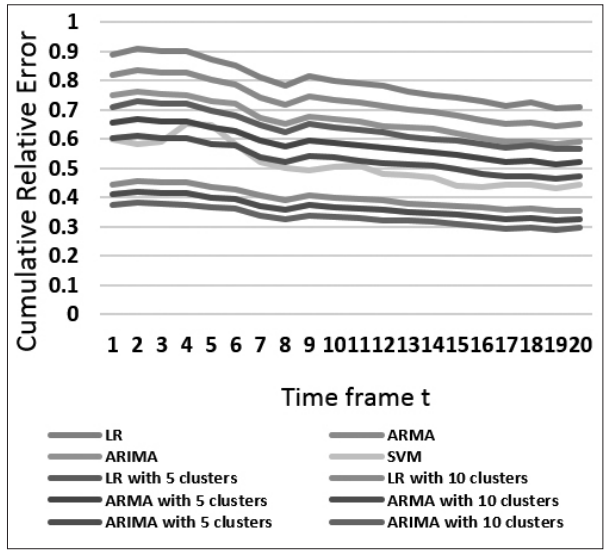
در این تحقیق محیط ابری در شبیه‌ســازی کلودســیم پیاده‌سازی شده اســت. در این شبیه‌سازی ده مرکز داده در نظر گرفته شــده اســت کــه ۳ مرکــز داده به‌صورت خصوصــی و ۷ مرکــز داده به‌صورت عمومــی تعریف شده اســت. مراکز داده خصوصی دارای محدودیت کمّی برای اعطای منابع به یک مشــتری می‌باشند و مراکز داده عمومی می‌توانند منابع را با محدودیت کمتری به مشتریان اعطاء نمایند. منابع از طریق ماشین‌های مجازی به‌کاربران اعطاء می‌گردد. خصوصیات ماشین‌های مجازی در مرکز داده‌هــای عمومی بر اســاس خصوصیت سیســتم ابری شرکت آمازون[9] طراحی شده است. سازوکار اعطای منابع به‌صورت پویا[10] تعریف شده است و وظایف براساس نیاز خود منابع آن را درخواســت نموده و محیط ابری پس از بررســی آن را به وظیفه اختصــاص می‌دهد. پارامترهای کلیدی توسط سیستم پایش سیستم ابری مشاهده شده و ســپس با اســتفاده از مدل پیش‌بینی ارائه شده تخمین‌زده می‌شــود. نتایج در اختیار مدل اقتصــادی قرار می‌گیرد تا بر اســاس سیاست‌های اقتصادی محیط ابری منابع مورد نیاز جذب و منابع مازاد آزادســازی شــوند و در صورت نیاز ســایر خوشه‌ها در اختیار آن‌ها قرار گیرد. وظایف بر اساس دو پارامتر کلیدی(اندازه وظیفه، تعداد پردازنده‌های مورد نیاز) خوشه‌بندی می‌شوند. وظایف در هر خوشه از لحاظ ساختاری تقریبا مشابه یکدیگر عمل می‌کنند بنابراین مدل پیش‌بینی می‌تواند بر روی هر خوشه مجزا عمل نماید. به این ترتیب مدل پیش‌بینی به‌صورت تطبیق‌پذیر بر روی شرایط یک خوشــه عمل خواهد کرد و تصمیماتی که مدل ارائه شده می‌تواند برای هر خوشه بگیرد مستقل از سایر خوشه‌ها است.

با توجه به این‌که اعطای منابع در محیط ابری به‌صورت اختصاصــی به یــک کاربر خاص غیر ممکن اســت، منابع به‌صورت اشــتراکی در میان کاربران توزیع می‌شود. برای حل این مشکل به هر کاربر تعداد محدودی از سیکل یک منبع اختصاص داده می‌شود. در نتیجه یک منبع بر اساس سیکل اشــتراکی[11] بین کاربران توزیع می‌شود. به اشتراک‌گذاری منابع تحت کنترل مدل ارائه شده خواهد بود به طوری‌که اگر اشــتراک‌گذاری سیستم بهنگام را در زمان اجرا به مخاطره بیندازد، سیســتم اشتراک‌گذاری ســیکل زمانی را متوقف خواهــد کرد. تصمیــم به توقف و یا ادامه اشــتراک گذاری ســیکل زمانی محدود به یک خوشــه خواهد بود. در نتیجه سایر خوشه‌ها می‌توانند بدون مشکل به‌کار خود ادامه دهند. هدف گذاری در این مدل به نحوی بوده است که منابع قرار اســت بین یک میلیون کاربر که قصد دارند از این سیســتم به‌صورت بهنگام استفاده کنند توزیع شود.

۶ -نتایج

در ایــن بخش پنــج آزمایش در محیط ابــری ترکیبی

9- Amazon EC2
10- Dynamic
11- Cycle sharing



شکل ۳: میزان خطای نسبی تجمعی در مدل‌های مختلف در ۲۰ قاب زمانی

صورت گرفته است و بر اساس تکنیک‌های ارزیابی کارآیی، عملکرد و دقت مدل ارائه شده را با سایر مدل‌های ارائه شده در بخش کارهای مرتبط مقایسه می‌کند.

۶-۱- تاثیر خوشه‌بندی بر روی میزان خطای نسبی تجمعی

در این آزمایش تمامی مدل‌هایی که در کارهای مرتبط این مقاله ارائه گردیده مورد آزمایش و مقایسه قرار گرفته است. در این آزمایش سعی شده میزان خطای نسبی تجمعی را برای تمامی مدل‌ها مورد بررسی قرار دهد. تمامی مدل‌ها بدون اشتراک سیکل زمانی مورد مقایسه قرار گرفته‌اند که در شکل ۳ قابل رویت می‌باشد. بنابراین مقدار کمتر ε نشان دهنده عملکرد بهتر سیستم و میزان دقت بالاتر آن‌ها در زمان اجرای وظایف بهنگام را نشان می‌دهد. داده‌ها از لحاظ زمانی به ۲۰ قاب زمانی تقسیم شده‌اند و میزان خطای نسبی تجمعی در هر قاب زمانی محاسبه شده و در شکل ۳ نشان داده شده است. در این آزمایش مدل‌های ARMA, AR و مدل ARIMA با و بدون خوشه‌بندی مورد استفاده قرار گرفته‌اند و با مدل SVMGA نیز مقایسه شده‌اند. نتایج نشان می‌دهد مدل‌هایی که از خوشه‌بندی استفاده کرده‌اند دارای نتایج بهتری نسبت به مدل‌هایی که از خوشه‌بندی استفاده نکرده بودند بوده است. به‌علاوه مدل SVMGA در مقایسه با مدل‌های ارائه شده که از ترکیب یک سری ساده زمانی و خوشه‌بندی به‌وجود آمده اند دارای میزان خطای بیشتری می‌باشند ولی در مقایسه با مدل‌هایی که از سری زمانی ساده و بدون خوشه‌بندی بوده‌اند بهتر عمل کرده است.

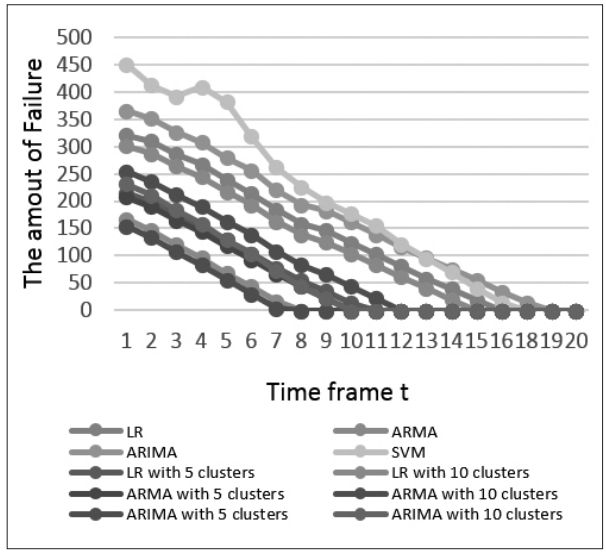
۶-۲ تاثیر خوشه‌بندی بر تعداد شکست‌های سیستم

در شکل ٤ وظایف مربوط به مجموعه داده‌های محیط ابری گوگل برای مدل‌های مرتبط اجرا گردیده است. سیستم در این آزمایش به مدل‌ها اجازه داده است که درصورتی‌که وظایف نیاز به دریافت منابع دارند آن منابع را دریافت نمایند و در صورتی‌که مدل تشخیص داد منابع موجود را می‌تواند بدون بروز مشکل بین کاربران بیشتری به اشتراک بگذارد این کار را انجام دهد. نتایج نشان می‌دهد که خوشه‌بندی تاثیر مثبتی بر روی کاهش نرخ شکست دارد.

در شکل ٤، بعد از ۲۰ قاب زمانی، در همه مدل‌ها با توجه به تطبیقی که در طی زمان مدل داشته، تعداد شکست‌ها بعد از طی چند دوره زمانی به سمت صفر میل کرده‌اند. همه مدل‌هایی که از خوشه‌بندی استفاده کرده‌اند توانسته‌اند با تجربه‌ای که از قاب‌های زمانی گذشته به‌دست آورده‌اند تعداد شکست‌های کمتری نسبت به سایر مدل‌ها داشته باشند و در مجموع بهتر عمل نموده‌اند. مدل SVMGA در شکل فوق با توجه به ساختار غیرتطبیقی و غیربازخورددار در یک قاب نتوانسته نسبت به سایر مدل‌ها موفق عمل نماید. اما در صورتی‌که سیستم بهنگام از نوع سخت در نظر گرفته شده باشد هیچکدام از مدل‌ها نمی‌توانند تضمین دهند که بدون فروپاشی سیستم به‌کار خود پایان دهند.

۶-۳ تاثیر به اشتراک‌گذاری منابع به همراه جذب منابع بیشتر بر روی میزان به‌کارگیری پردازنده‌ها

در این آزمایش که در شکل ٥ دیده می‌شود، سیستم به‌صورت تطبیقی برای هر خوشه بر اساس نیاز خوشه شروع به جذب منابع بیشتر و به اشتراک‌گذاری آن بین



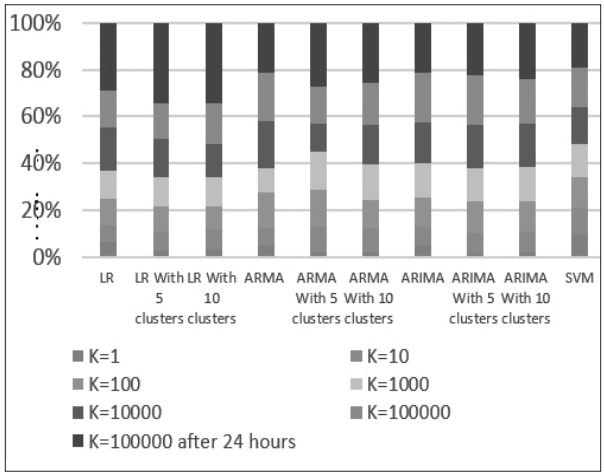
شکل ۴ : میزان شکست در مدل‌های مختلف

تعداد بیشـــتری از کاربران می‌نماید. به زبان ســـاده‌تر در این راهبرد، سیســـتم منابع بیشتری به‌دست می‌آورد ولی با به اشتراک گذاشتن آن‌ها بین تعداد بیشتری از مشتریان هزینه ســـربار آن منابع را بی‌تاثیر می‌کند و یا حتی کاهش می‌دهد. میزان به‌کارگیری منابع در شکل ۵ نشان داده شده است. نتایج نشان می‌دهد که سیستم توانسته به خوبی با به اشتراک‌گذاری منابع، منابعی که بیکار بوده‌اند را در اختیار سایر کاربران قرار دهند.

سیســـتم با ایجاد کنترل بر روی میزان اشتراک‌گذاری ســـیکل زمانـــی و میزان جذب منبع جدید، ســـعی در حفظ نســـبت بین ایـــن دو کرده و مانع از جذب بـــی رویه منابع اضافی گردد. فقط در صورتی منبع اضافی توسط این مدل جذب می‌شـــود که سیســـتم بتواند آن منابع را با کاربران دیگر به اشـــتراک بگذارد. در شکل ۵ می‌توان دید که میزان به‌کارگیری منابع در زمانی‌که منابع با کاربران بیشـــتری به اشـــتراک گذاشته شده‌اند با وجود این‌که منابع بیشتری جذب شده‌اند بیشتر بوده است.

۶-٤- اندازه‌گیری میزان بهره وری مدل‌ها با استفاده از خطای مجذور متوسط ریشه

در شـــکل ٦ می‌توان دید که تمامی مدل‌های ارائه شده در کارهـــای مرتبط از لحاظ بهره‌وری با یکدیگر مقایســـه

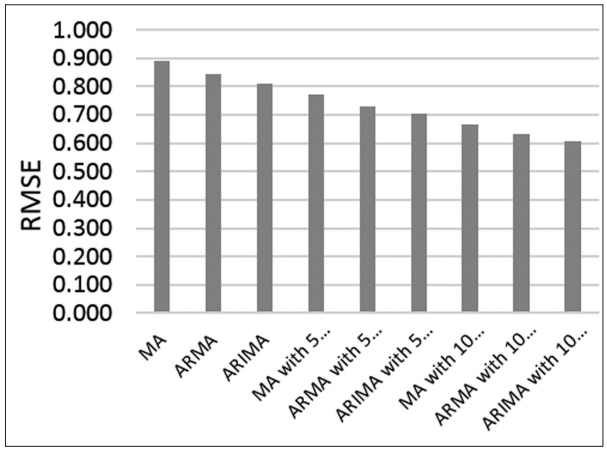


شکل ۵: میزان به‌کارگیری منابع وقتی منابع بین کاربران به اشتراک گذاشته می‌شود

شـــده‌اند و مقدار خطای مجذور ریشـــه بـــرای همه آن‌ها محاسبه شده اســـت. همان‌طور که در بخش ٤-۲-۲ گفته شـــد هر چقدر مقـــدار این متغیر به صفر نزدیک‌تر باشـــد مدل دارای بهره‌وری بالاتری می‌باشـــد. نتایج اندازه‌گیری نشـــان می‌دهد مدل‌های تطبیقی که از خوشه‌های بیشتری بـــرای کار خود بهره گرفته‌اند از میزان بهره وری بالاتری برخـــوردار بوده‌اند ولی مدل SVM کـــه فاقد تطبیق‌پذیری بوده اســـت در میـــان مدل‌ها از ضعیف‌تریـــن مقدار بهره وری برخوردار می‌باشـــد. می‌توان نتیجه گرفت که اتخاذ تصمیم‌های متفاوت برای هر خوشـــه در هر دوره زمانی مشخص توانســـته مدل‌هایی که از خوشـــه‌بندی استفاده نموده‌اند را از لحاظ میزان بهره وری بهبود بخشد.

۶-۵ اندازه‌گیری میزان دقت و شکست مدل‌ها

در شـــکل ۷ میزان R2 برای تمامی مدل‌ها اندازه‌گیری شده است تا میزان دقت مدل‌ها محاسبه گردد. همان‌طور که در بخش بهره‌وری از متغیرهای اندازه‌گیری عملکرد گفته شـــد در صورتی‌که مقدار R2 در یک مدل به یک نزدیک‌تر باشد آن مدل از دقت بیشتری برخوردار می‌باشد. به‌علاوه میزان خطا نیز برای همه مدل‌ها اندازه‌گیری شـــده است تا بتواند همبســـتگی بین دقت ومیزان خطا را نشـــان دهد. بر اساس نتایج نمایش داده شده در شکل ۷ نزدیک‌ترین مقدار به صفر برای میزان خطا متعلق به دقیق‌ترین مدل می‌باشد.

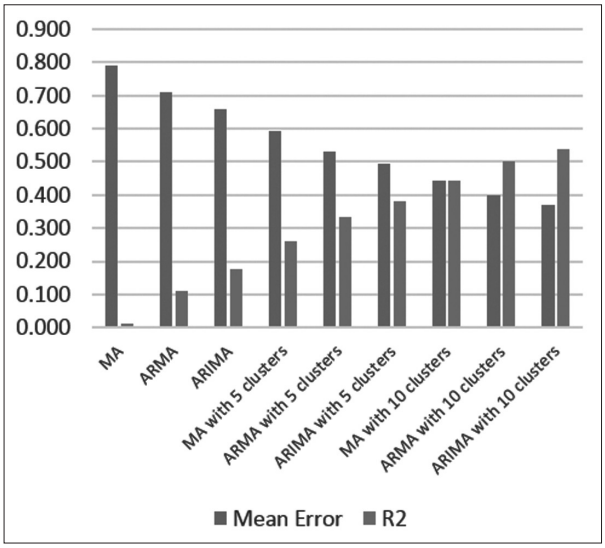


شکل ۶: اندازه‌گیری میزان بهره وری مدل‌های ارائه شده

به‌علاوه بر اســـاس اطلاعاتی که از شکل ۷ به‌دست می‌آید تمامی مدل‌هایی که از خوشه‌بندی استفاده کرده‌اند از میزان خطای متوســـط کمتری برخوردار هستند. همچنین میزان دقت آن‌ها نیز در مقایســـه با مدل‌هایی که از خوشـــه‌بندی استفاده نکرده‌اند بیشتر می‌باشد.

۷- نتیجه گیری

در این مطالعه بر اســـاس مدلی که ارائه گردیده سعی شـــده تا ناهنجاری احتمالی به‌صورت خودکار تشـــخیص داده شـــود. این ناهنجاری‌ها که شامل کم‌کاری، بیکاری و یا ازدحام در بعضی از منابع می‌شود مبنای تصمیم‌گیری در این مدل می‌باشد. هدف اصلی این تحقیق ارائه یک مدل پیش‌بینـــی برای تضمین اجرای وظایـــف بهنگام در ظرف مهلت زمانی مقرر است که به‌طور کامل محقق نشده است. در نتایج حاصل شـــده در میزان به‌کارگیری منابع و دقت میزان منابعی که باید تخصیص یابد نتایج زیر حاصل شد. اولین نتیجه نشان می‌دهد که افزونه بهنگام در تشخیص در زمانی‌که میزان درخواست‌های منابع کم می‌باشد با مشکل روبرو است و منابع بیشتر از نیاز جذب می‌نماید. اگر چه، این افزونه قابلیت تشخیص و تخمین میزان مناسب از منابع را در زمان عادی بودن تعداد درخواست‌ها برای منابع دارا می‌باشد. بازخورد در مدل پیش‌بینی نشان داده که سیستم پس از گذشت دوره‌های زمانی از میزان خطای خود کاسته



شکل ۷: میزان دقت و متوسط خطا

و هرچه افزونه به‌صورت طولانی‌تر به‌کار گرفته می‌شود از میزان خطای کمتری برخوردار شـــده اســـت. در نتیجه سیستم توانسته با اســـتفاده از بازخورد خطای پیش‌بینی خـــود را اصلاح نماید. در نهایت میـــزان تطبیق‌پذیری مدل ارائـــه شـــده در نمونه‌هایی که از خوشـــه‌بندی اســـتفاده نموده‌اند بهبود را نشان می‌دهد.

سیستم ابری می‌تواند با وضعیت رخ داده خود را تطبیق دهد و می‌تواند به‌کاربران به‌عنوان توافقنامه سطح خدمات برای اجرای وظایف بهنگام تحت عنوان یک سرویس ارائه گردد. افزونه ارائه شده می‌تواند برای سیستم‌های بهنگام نرم مناسب عمل کرده و با میزان کم تخلف از مهلت زمانی وظایـــف را اجرا نماید. این مـــدل می‌تواند منابع را به کمک مدل اقتصادی مناســـب از تامین‌کنندگان منابع خصوصی و عمومی مرتفع نماید که در تحقیقات آتی مورد بررســـی قرار می‌گیرد.

۸- مراجع

[1] K. G. Shin and P. Ramanathan, “Real-time computing: a new discipline of computer science and engineering,” Proc. IEEE, vol. 82, no. 1, pp. 6–24, 1994.

[2] C.-J. Huang, C.-T. Guan, H.-M. Chen, Y.-W. Wang, S.-C. Chang, C.-Y. Li, and C.-H. Weng, “An adaptive resource management scheme in cloud computing,” Eng. Appl. Artif. Intell., vol. 26, no. 1, pp. 382–389, 2013.

[3] H. Moreira, Orlando and Corporaal, chakraborty2012advanc-

and computing resource in private cloud," Comput. Ind., vol. 64, no. 4, pp. 448–463, 2013.
[22] A. Hameed, A. Khoshkbarforoushha, R. Ranjan, P. P. Jayaraman, J. Kolodziej, P. Balaji, S. Zeadally, Q. M. Malluhi, N. Tziritas, A. Vishnu, S. U. Khan, and A. Zomaya, "A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems," Computing, Jun. 2014.
[23] S.-R. Kuang, K.-Y. Wu, B.-C. Ke, J.-H. Yeh, and H.-Y. Jheng, "Efficient architecture and hardware implementation of hybrid fuzzy-Kalman filter for workload prediction," Integr. VLSI J., vol. 47, no. 4, pp. 408–416, Sep. 2014.
[24] Y. S. Sun, Y.-F. Chen, and M. C. Chen, "A Workload Analysis of Live Event Broadcast Service in Cloud," Procedia Comput. Sci., vol. 19, no. Fams, pp. 1028–1033, 2013.
[25] A. Gregoriades and A. Sutcliffe, "Workload prediction for improved design and reliability of complex systems," Reliab. Eng. Syst. Saf., vol. 93, no. 4, pp. 530–549, Apr. 2008.
[26] M. Calzarossa and G. Serazzi, "Workload characterization: A survey," Proc. IEEE, vol. 81, no. 8, pp. 1136–1150, 1993.
[27] G. Kousiouris, A. Menychtas, D. Kyriazis, S. Gogouvitis, and T. Varvarigou, "Dynamic, behavioral-based estimation of resource provisioning based on high-level application terms in Cloud platforms," Futur. Gener. Comput. Syst., vol. 32, pp. 27–40, 2014.
[28] D. Hutchison, Network and Parallel Computing. Springer Heidelberg NewYork Dordrecht London, 2014.
[29] S. Mallick, A. Einstein, C. Marne, and C. S. Deme, "A Resource Prediction Model for Virtualization Servers," pp. 667–671, 2012.
[30] Y. Jiang, C. S. Perng, T. Li, and R. Chang, "ASAP: A self-adaptive prediction system for instant cloud resource demand provisioning," Proc. - IEEE Int. Conf. Data Mining, ICDM, pp. 1104–1109, 2011.
[31] G. Kryftis, Yiannos and Mavromoustakis, Constandinos X and Mastorakis, George and Pallis, Evangelos and Batalla, Jordi Mongay and Skourletopoulos, "Resource usage prediction for optimal and balanced provision of multimedia services," in Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2014 IEEE 19th International Workshop on, 2014, pp. 255–259.
[32] N. Doulamis, A. Doulamis, A. Litke, A. Panagakis, T. Varvarigou, and E. Varvarigos, "Adjusted fair scheduling and nonlinear workload prediction for QoS guarantees in grid computing," Comput. Commun., vol. 30, no. 3, pp. 499–515, Feb. 2007.
[33] Z. Gong, X. Gu, and J. Wilkes, "Press: Predictive elastic resource scaling for cloud systems," in Network and Service Management (CNSM), 2010 International Conference on, 2010, pp. 9–16.
[34] P. Kantor, L. Rokach, F. Ricci, and B. Shapira, Recommender systems handbook. Springer, 2011.
[35] G. Zhu, Qian and Agrawal, "Resource provisioning with budget constraints for adaptive applications in cloud environments," 2010, pp. 304–307.
[36] M. A. Reiss, Charles and Tumanov, Alexey and Ganger, Gregory R and Katz, Randy H and Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," 2012, p. 7.
es. Springer, 2014.
[4] S. Islam, J. Keung, K. Lee, and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," Futur. Gener. Comput. Syst., vol. 28, no. 1, pp. 155–162, 2012.
[5] J. Liu, Chunhong and Liu, Chuanchang and Shang, Yanlei and Chen, Shiping and Cheng, Bo and Chen, "An adaptive prediction approach based on workload pattern discrimination in the cloud," J. Netw. Comput. Appl., vol. 80, pp. 35–44, 2017.
[6] L. Barros, P. López Martínez, C. Cuevas, J. M. Drake, and M. González Harbour, "Modelling real-time applications based on resource reservations," J. Syst. Archit., vol. 59, no. 6, pp. 315–330, 2013.
[7] M. Lin, Z. Yao, and T. Huang, "A hybrid push protocol for resource monitoring in cloud computing platforms," Opt. - Int. J. Light Electron Opt., pp. 1–5, 2015.
[8] R. Weingärtner, G. B. Bräscher, and C. B. Westphall, "Cloud resource management: A survey on forecasting and profiling models," J. Netw. Comput. Appl., vol. 47, pp. 99–106, Jan. 2015.
[9] B. Javadi, J. Abawajy, and R. Buyya, "Failure-aware resource provisioning for hybrid Cloud infrastructure," J. Parallel Distrib. Comput., vol. 72, no. 10, pp. 1318–1331, 2012.
[10] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger, "Economics Paradigm for Resource Management and Scheduling in Grid Computing," Concurr. Comput. Pract. Exp., vol. 14, pp. 1507–1542, 2002.
[11] R. Pal and P. Hui, "Economic models for cloud service markets: Pricing and Capacity planning," Theor. Comput. Sci., vol. 496, no. 2013, pp. 113–124, 2013.
[12] D. Kusic and N. Kandasamy, "Risk-aware limited lookahead control for dynamic resource provisioning in enterprise computing systems," Cluster Comput., vol. 10, no. 4, pp. 395–408, Aug. 2007.
[13] S. and E. Chakraborty, Advances in Real-Time Systems. Springer Science, 2012.
[14] J. Malawski, Maciej and Juve, Gideon and Deelman, Ewa and Nabrzyski, "Algorithms for cost-and deadline-constrained provisioning for scientific workflow ensembles in IaaS clouds," Futur. Gener. Comput. Syst., vol. 48, pp. 1–18, 2015.
[15] B. Javadi, P. Thulasiraman, and R. Buyya, "Enhancing performance of failure-prone clusters by adaptive provisioning of cloud resources," J. Supercomput., vol. 63, pp. 467–489, 2013.
[16] A. R. Systems, Advances in Real-Time Systems. Springer Heidelberg Dordrecht London New York, 2012.
[17] H. Kopetz, Real-Time Systems: Design Principles for Distributed Embedded Applications. 2011.
[18] H. Moreira, Orlando and Corporaal, Scheduling real-time streaming applications onto an embedded multiprocessor. Springer, 2014.
[19] K. Eisenreich, G. Hackenbroich, V. Markl, and R. Philipp, "Handling of Uncertainty and Temporal Indeterminacy for What-if Analysis," pp. 100–115, 2011.
[20] J. Tong, H. E, M. Song, and J. Song, "Host load prediction in cloud based on classification methods," J. China Univ. Posts Telecommun., vol. 21, no. 4, pp. 40–46, Aug. 2014.
[21] Y. Laili, F. Tao, L. Zhang, Y. Cheng, Y. Luo, and B. R. Sarker, "A Ranking Chaos Algorithm for dual scheduling of cloud service