نشریه علمی «پدافند الکترونیکی و سایبری»
سال هشتم، شماره ۲، تابستان ۱۳۹۹؛ ص ۱۰۳- ۹۵

علمی–پژوهشی

ارائه یک روش زمانبندی وظیفه تحمل‌پذیر خطا به منظور استفاده بهینه ازمنابع در محیط رایانش ابری

محمدجواد احراری[۱]، محمدرضا حسنی‌آهنگر[۲*]، آرش غفوری[۳]
۱- کارشناسی ارشد، ۲- استاد، ۳- پژوهشگر دانشگاه جامع امام حسین(ع)
(دریافت: ۹۸/۵/۱۴، پذیرش: ۹۸/۸/۱)

چکیده

در سال‌های اخیر رایانش ابری در حال تبدیل شدن به یک فناوری مهم در حوزه‌ی فناوری اطلاعات است. در محیط رایانش ابری احتمال بروز خطا وجود دارد. روش‌های متفاوتی برای مقابله با خطاها وجود دارد ولی با توجه به ویژگی‌ها و خصوصیت‌های محیط رایانش ابری، استفاده از روش‌های تحمل‌پذیری خطا بهترین انتخاب برای مقابله با خطا در این محیط است. یکی از بزرگ‌ترین مسئله‌ها در روش‌های تحمل‌پذیری خطا، استفاده بهینه از منابع است. استفاده بهینه از منابع هم برای فرآهم‌آورندگان سرویس‌های ابری و هم برای مشتریان سرویس‌های ابری دارای اهمیت زیادی است. متأسفانه استفاده بهینه از منابع در روش‌های تحمل‌پذیری خطا در رایانش ابری، خیلی مورد توجه پژوهشگران و فرآهم آورندگان سرویس‌های ابری قرار نگرفته است. در این مقاله سعی شده است با در نظرگرفتن وابستگی بین وظایف، یک روش تحمل‌پذیری خطا بر روی ماشین‌های مجازی ارائه شود که علاوه بر تحمل‌پذیر بودن در برابر خطا، به بهینگی در استفاده از منابع نیز دست یابد. در این روش با استفاده از یک زمانبند اولویت‌دار، به هر یک از وظایف یک اولویت اختصاص داده می‌شود. سپس وظایف به ترتیب اولویت‌شان جهت پردازش به ماشین‌های مجازی فرستاده می‌شوند. نتایج حاصل از شبیه‌سازی روش پیشنهادی توسط شبیه‌ساز کلاودسیم نشان می‌دهد که روش پیشنهادی نسبت به روش تکرار توانسته است بهینگی بیشتری در استفاده از منابع را به ارمغان بیاورد و با ضریب اطمینان ۹۵ درصد، به ۲۹/۱۵٪ و ۲۲/۷۴٪ بهبود در استفاده از تعداد پردازنده و ۳۰/۷۶٪ و ۲۲,۳۴٪ بهبود در استفاده از حافظه و ۲۹/۷۱٪ و ۲۲/۸۸٪ بهبود در استفاده از پهنای باند دست یافته است.

کلید واژه‌ها: رایانش ابری، تحمل‌پذیری خطا، ماشین مجازی، بهینگی منابع، زمانبندی وظایف

۱- مقدمه

امروزه رایانش ابری شهرت بسیار زیادی پیداکرده و نمونه‌ای از محاسبات توزیع‌شده، متشکل از منابع و درخواست‌ها باهدف اشتراک‌گذاری منابع به‌صورت سرویس در بستر اینترنت است. این محیط تصوری از منابع بی‌نهایت را برای کاربران فراهم می‌کند؛ بنابراین کاربران می‌توانند مبتنی بر درخواست، میزان استفاده از منابعشان را افزایش یا کاهش دهند. تحمل‌پذیری خطا یکی از نگرانی‌های عمده برای تضمین در دسترس بودن و قابلیت اطمینان خدمات حیاتی و هم‌چنین اجرای برنامه‌های کاربردی است. به‌منظور به حداقل رساندن تأثیر خرابی بر روی سامانه و اجرای صحیح و موفق برنامه‌های کاربردی، خرابی‌ها باید پیش‌بینی‌شده و فعالانه مدیریت و کنترل گردند. روش‌های تحمل‌پذیری خطا درواقع برای پیش‌بینی این خرابی‌ها و انجام یک اقدام مناسب قبل از وقوع خرابی است [۱]. اگرچه تحمل‌پذیری خطا در سامانه‌های توزیع‌شده موضوعی درگذشته بوده است که مجموعه وسیعی از الگوریتم‌ها برای تشخیص، شناسایی و اصلاح خطا ایجاد کرده است اما می‌بایست این مفاهیم را درزمینه رایانش ابری دوباره مورد مطالعه قرارداد. یکی از مهمترین چالش‌های روش‌های تحمل‌پذیری خطاهای موجود، استفاده بهینه و کارآمد از منابع است که متاسفانه خیلی مورد توجه پژوهشگران و فرآهم آورندگان سرویس‌های ابری قرار نگرفته است. در ابر، فراهم‌کنندگان می‌خواهند بیش‌ترین بازده را از منابع خود ببرند و کاربران نیز می‌خواهند هزینه‌های خود را حداقل نمایند درعین‌حال، عملکرد مورد نیازشان را نیز به دست آورند.

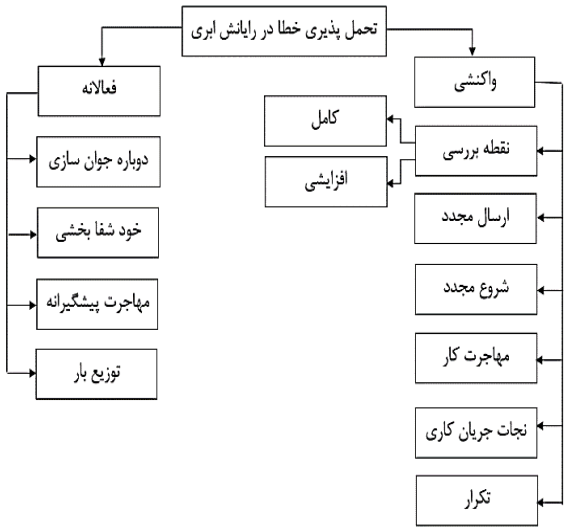
استفاده مناسب و بهینه از منابعی همچون حافظه، پردازشگر و سایر منابع محاسباتی یک چالش است ازاین‌رو، چگونگی زمانبندی وظایف، مسئله‌ای مهم محسوب می‌شود که تأثیر زیادی در عملکرد فراهم‌کنندگان سرویس ابر دارد. زمان‌بندی وظایف،

*رایانامه نویسنده پاسخگو mrhasani@ihu.ac.ir

انتخاب بهترین منبع مناسب با هدف انتشار بار در پردازنده‌ها و حداکثر بهره‌وری از منابع است درحالی‌که باید زمان پاسخ و تکمیل هر وظیفه و همچنین هزینه سرویس را حداقل نماید [۱]. روش‌های زمانبندی متفاوتی برای اختصاص منابع به وظایف و برنامه‌های کاربردی در حال اجرا وجود دارد که هرکدام از این روش‌ها سعی در بهینه‌سازی استفاده از منابع محاسباتی و افزایش کارایی سامانه را دارند. بنابراین، با توجه به اهمیت استفاده بهینه از منابع، در این مقاله یک روش زمانبندی وظیفه تحمل‌پذیر خطا ارائه شده است که علاوه بر تحمل‌پذیر بودن در برابر خطاها، با استفاده از یک الگوریتم زمان‌بندی وظایف، استفاده بهینه از منابع را نیز به ارمغان می‌آورد.

۱-۱- تحمل‌پذیری خطا در رایانش ابری

عموماً یکی از راه های مقابله با خطاها جلوگیری از به‌وجود آمدن خطاها است. با توجه به معماری پیچیده رایانش ابری و وجود مراکز داده بسیار بزرگ و ارتباطات زیاد بین آن‌ها استفاده از روش‌های جلوگیری از خطا در فاز توسعه معقول نیست و کارایی مناسب را ندارد. روش‌های جلوگیری از خطا مانند آزمایش کمک به تشخیص و از بین بردن خطاها می‌کند، ولی در مورد رایانش ابری کارایی لازم را ندارد. بنابراین، برای افزایش قابلیت اطمینان و دسترس‌پذیری باید سامانه تحمل‌پذیر خطا باشد. عواملی مانند بازیابی خرابی، هزینه کمتر، افزایش کارایی دلایل دیگری برای استفاده از تحمل‌پذیری خطا در زیرساخت رایانش ابری است. فن‌های تحمل‌پذیری خطا در رایانش ابری بر اساس سیاست‌های استفاده شده در دو دسته تحمل‌پذیری خطای واکنشی و تحمل‌پذیری خطای فعالانه قرار می‌گیرند [۳-۲]. شکل (۱) فن‌های تحمل‌پذیری خطا در رایانش ابری را نشان می‌دهد.



شکل (۱): فن‌های تحمل‌پذیری خطا در رایانش ابری.

کارهای انجام شده در زمینه تحمل‌پذیری خطا در رایانش ابری با توجه به نیازهای خود از این روش ها به منظور نیل به اهدف خود استفاده می‌نمایند.

۲- کارهای مرتبط

۱- پلانکستینر و همکارانش روشی به نام ارسال مجدد وظیفه را معرفی کردند. در این روش، وظیفه‌ای که دچار خرابی شده است برای اجرای دوباره آن وظیفه، دو راه حل وجود دارد. در راه حل اول وظیفه خراب شده برای اجرای دوباره به همان منابع قبلی ارسال می شود. درصورتی که خرابی وظیفه به دلیل مشکلات سخت‌افزاری باشد، وظیفه دوباره با خطا مواجه می شود و در یک حلقه بی پایان قرار می‌گیرد و وظیفه هیچ‌گاه با موفقیت به پایان نمی‌رسد. در راه حل دوم وظیفه خراب شده برای اجرای دوباره به منابع جایگزین ارسال می شود که این کار باعث استفاده غیر بهینه از منابع می‌شود. درواقع در این روش هیچ سازوکاری برای تعیین نوع خطا به‌کار گرفته نشده است و مستقل از نوع خطا یکی از دو روش ذکرشده به‌کار گرفته می‌شود [۴].

۲- نگاشت/کاهش یک چارچوبی برای پردازش مجموعه‌های عظیمی از داده‌ها بر روی گره‌ها است که بر روی موضوعی خاص فعالیت می‌کنند. این مجموعه روی‌هم رفته به‌عنوان یک خوشه شناخته می‌شود (در صورتی که از سخت‌افزاری یکسان بهره برند). پردازش محاسباتی بر روی دادهایِ ذخیره‌شده درون فایل (ساختار نیافته) یا بر روی پایگاه داده (ساختاریافته) قابل اجراست که برای انجام برنامه‌های کاربردی با مجموعه داده بسیار بزرگ سروکار دارد. در گامِ نگاشت گره اصلی ورودی را به قطعاتی کوچک‌تر تقسیم می‌نماید (تقسیم مسئلهٔ بزرگ به مسائل کوچک) و سپس این مسائل کوچک را به‌عنوان ورودی به گره‌های کارگر ارائه می‌دهد. هر گره کارگر زیرمسئله خود را حل نموده و نتیجه را به گره اصلیِ خود برمی‌گرداند. در گامِ کاهش سپس گرهِ اصلی جواب زیرمسایل را از گره‌های کارگرش گرفته و خروجی را می‌سازد. خرابی یک گره در این سامانه باعث عدم تکمیل کار می‌شود و در نتیجه نتایج گره‌های دیگر بی‌اثر می‌ماند و باعث اتلاف زیاد هزینه زمانی و منابع می‌شود. برای جلوگیری از این‌گونه هزینه‌ها، از تحمل‌پذیری خطا به‌وسیله انتقال کارهای قطع‌شده در ماشین مجازی خراب به ماشین مجازی سالم و در دسترس پشتیبانی می‌کند یعنی دیگر ماشین مجازی خراب شده تأثیری بر ماشین‌های مجازی دیگر نداشته و یک ماشین مجازی دیگر جایگزین ماشین مجازی خراب شده می‌شود. همچنین در این مدل برای جلوگیری از اجرای دوباره کارهای خراب‌شده از ابتدا، از روش نقطه بررسی استفاده شده است تا حالات کار جاری در دوره‌های ثابت ذخیره شود تا در صورت بروز خرابی در کار،

کار خراب‌شده از ابتدا اجرا نشود و از آخرین نقطه بررسی شروع شود تا زمان انجام کار طولانی نگردد [۵].

از معایب این روش سربار زیاد ناشی از ذخیره و نگهداری حالات وظایف و برنامه های کاربردی در حال اجرا است. هم‌چنین احتمال وقوع خطا در هنگام ذخیره حالات سامانه از دیگر چالش‌های این روش است که باید موردتوجه قرار گیرد. در این روش وظایفی که دچار خطا شده‌اند به‌طور کامل به ماشین مجازی دیگر انتقال می‌یابند و هیچ سازوکاری برای استفاده بهینه از منابع در نظر گرفته نشده است.

۳- سرویس های بی‌درنگ از جمله سرویس هایی هستند که در رایانش ابری همواره مورد استفاده قرار می گیرند. سامانه‌های بی درنگ با داشتن امتیازاتی از قبیل ابزارهای محاسبات قوی، محیط مجازی‌سازی با مقیاس‌پذیری بالا از اهمیت به‌سزایی برخوردارند. در رایانش ابری استفاده از گره‌های پردازشگر احتمال اشکال را بالا می‌برد و از طرفی چون سامانه‌های بی‌درنگ از لحاظ ایمنی بحرانی هستند باید قابلیت اطمینان آن‌ها افزایش یابد از این‌رو، تقاضا برای دست‌یابی به سامانه‌هایی با قابلیت تحمل خطا در سامانه‌های بی‌درنگ رو به افزایش است. با توجه به اهمیت بالای سرویس‌های ابری، مالک و همکارانش یک روشی برای مدیریت خطاها و تصمیم‌گیری بر اساس قابلیت اطمینان ماشین های مجازی برای سرویس‌های ابری ارائه کرده‌اند. در این مدل قابلیت اطمینان ماشین‌های مجازی در هر چرخه محاسباتی تغییر می‌کند. در هر چرخه اگر ماشین مجازی نتایج درست تولید کند و این نتایج را در مدت زمان مشخص ارائه دهد، قابلیت اطمینان آن ماشین مجازی افزایش می‌یابد ولی درصورتی که نتایج درست تولید نکند یا نتایج درست تولید کند ولی در مدت زمان تعیین شده، نتواند ارائه دهد، در این صورت قابلیت اطمینان آن ماشین مجازی کاهش پیدا می کند. اگر قابلیت اطمینان ماشین مجازی به صورت مداوم کاهش یابد، آن ماشین مجازی حذف می‌شود و ماشین مجازی دیگری جایگزین آن می‌شود. قابلیت اطمینان تمامی ماشین‌های مجازی با حداقل سطح قابلیت اطمینان مورد بررسی قرار می‌گیرد و چنانچه حداقل سطح قابلیت اطمینان توسط ماشین مجازی به‌دست آید بنابراین، آن ماشین مجازی مناسب است و در غیر این از سازوکار بازیابی رو به عقب استفاده می‌کند [۶].

در این روش که بیشتر برای سرویس‌های بی درنگ در نظر گرفته شده است؛ فقط قابلیت اطمینان ماشین های مجازی مورد توجه قرارگرفته است و در صورتی که قابلیت اطمینان ماشین مجازی از یک حد آستانه کمتر شود، از چرخه کنار گذاشته می‌شود و ماشین مجازی جدید جایگزین آن می‌شود. در این روش هیچ راه کاری برای استفاده بهینه و مناسب از منابع در نظر گرفته نشده است و می‌توان گفت که این روش غیر بهینه‌ترین روش برای تحمل‌پذیری خطا در رایانش ابری است.

۴- مشرام و همکارانش با توجه به اهمیت سرویس‌های بی درنگ در رایانش ابری یک روش تحمل پذیری خطا در محیط رایانش ابری برای افزایش قابلیت اطمینان ارائه کرده‌اند. در این روش که بهبود یافته روش قبلی است الگوریتم موجود در هر ماشین مجازی برای انجام وظایف با الگوریتم موجود بر روی سایر ماشین های مجازی متفاوت است یعنی هر ماشین مجازی وظایف خود را به‌صورت متفاوت با سایر ماشین‌های مجازی انجام می‌دهد. در این روش قابلیت اطمینان ماشین‌های مجازی با توجه به نتایجی که تولید می‌کنند محاسبه می‌شود. مانند روش قبلی اگر ماشین مجازی نتایج درست در محدوده زمان تعیین‌شده تولید کند، قابلیت ماشین مجازی افزایش می‌یابد ولی اگر ماشین مجازی نتایج درست تولید نکند و یا نتایج درست تولید کند ولی نتایج را در محدوده زمان تعیین شده نتواند ارائه نماید قابلیت اطمینان ماشین مجازی کاهش پیدا می کند. قابلیت اطمینان هر ماشین مجازی در هر چرخه با حداقل قابلیت اطمینان قابل‌قبول مقایسه می‌شود چنانچه قابلیت اطمینان ماشین مجازی از حداقل قابلیت اطمینان قابل قبول کمتر شود آن ماشین مجازی از سامانه کنار گذاشته می‌شود و یک ماشین مجازی با قابلیت اطمینان قابل قبول جایگزین آن می‌شود. در این روش هم‌چنین از روش نقطه بررسی استفاده شده است تا در صورت بروز خطا در سامانه از بازیابی رو به عقب استفاده شود و زمان پاسخ‌گویی نسبت به روش قبل بهتر شود [۷].

این روش که بیشتر برای افزایش قابلیت اطمینان سرویس های بی‌درنگ در رایانش ابری به کار می‌رود تمامی تمرکز خود را بر اجرای سریع وظایف در محدوده زمانی تعیین شده، قرار داده است و در صورتی که ماشین مجازی نتواند در آن زمان وظیفه خود را انجام دهد و یا نتایج نادرست تولید کند در صورت پایین بودن سطح قابلیت اطمینان، از سامانه کنار گذاشته می‌شود. در اینجا باز استفاده بهینه از منابع درنظر گرفته نمی‌شود و سامانه تمامی تمرکز خود را بر روی قابلیت اطمینان و زمان پاسخ گذاشته است. ولی مزیت این روش نسبت به روش قبلی در بهتر بودن زمان پاسخگویی است.

۵- ژیامین ژاو و جی وانگ و همکارانش، یک مدلی برای تحمل‌پذیر خطای جریان کاری در رایانش ابری ارائه داده اند. یک جریان کاری مجموعه‌ای از وظایف است که اجرای آن‌ها به‌هم وابسته است یعنی خروجی یک وظیفه در اجرای یک وظیفه دیگر تأثیر دارد. این مدل، توسعه‌یافته روش پشتیبان‌گیری اصلی با

ترکیب ویژگی‌های ابری است. بر اساس این مدل، آن‌ها از یک روش برای اختصاص منابع به وظایف استفاده می‌کنند و از یک پیام به منظور اطمینان از این‌که خطاهایی که در حین اجرای جریان کاری به‌وجود می‌آیند و تحمل می‌شوند، استفاده می‌کنند. در آخر آن ها یک الگوریتم زمان‌بندی تحمل‌پذیر خطای پویا به نام *FASTER* برای جریان کاری ارائه دادند، که دارای سه ویژگی اصلی زیر است:

• از یک روش جابه‌جایی رو به عقب برای استفاده کامل از منابع استفاده می‌کند.

• از روش‌های افزایش مقیاس عمودی/افقی به‌منظور تأمین سریع منابع استفاده می‌کند.

• از روش کاهش مقیاس عمودی برای جلوگیری از تغییرات غیرضروری استفاده می‌کند [۸].

همان‌طور که در بالا اشاره شد، یکی از ویژگی های این روش تأمین سریع منابع برای اجرای هر چه سریع‌تر وظایف است؛ یعنی در صورت بروز خطا این روش به سرعت منابع جدید را جایگزین می‌کند تا وظایفی که دچار خطا شده‌اند را به‌سرعت زمان‌بندی کند و این موجب عدم استفاده بهینه از منابع می‌شود. در این روش افزایش سریع منابع باعث می شود تا در صورت بروز خطا، سرویس با تأخیر زمانی زیادی روبه‌رو نشود.

۶- ژائو و همکارانش، یک الگوریتم زمان‌بندی تحمل‌پذیر خطا برای وظایف موجود در رایانش ابری به نام *QAFT* ارائه داده‌اند. این روش به‌صورت ترکیبی از طرح‌های اشتراکی است که در یک خوشه قرار دارند. در این روش کیفیت خدمات ، قابلیت اطمینان، سازگاری و محدودیت زمانی برای انجام وظایف در رایانش ابری در نظر گرفته شده است. یعنی وظایف انجام شده با یک سطح کیفیت خدمات مقایسه می‌شوند تا سرویس‌هایی با کیفیت خدمات تضمین‌شده به کاربران ارائه دهند. در این روش سرویس‌های ارائه شده دارای قابلیت اطمینان بالا هستند و در بازه زمانی تعیین‌شده به اتمام می‌رسند. این الگوریتم در عین حال که قادر است تا خرابی ثابت یک گره را در یک لحظه آنی برای وظایف بی‌درنگ تحمل کند، نیازمندی‌های کیفیت خدمات وظایف را نیز برآورده کرده است. *QAFT* به‌منظور بهبود زمان‌بندی و قابلیت اطمینان تلاش می‌کند تا زمان شروع کپی‌های اصلی را جلو ببرد و در زمان شروع اجرای کپی‌های پشتیبان تأخیر بیندازد. هم‌چنین این الگوریتم برای بهبود انعطاف‌پذیری سعی می‌کند تا بین سطح کیفیت خدمات تضمین‌شده و اجرای کپی‌های پشتیبان تعادل ایجاد کند [۹].

این روش تمامی تلاش خود را در جهت ارائه سرویس ها با کیفیت خدمات تضمین شده و با قابلیت اطمینان بالا قرار داده است که ارائه سرویس با کیفیت خدمات تضمین شده در صورت بروز خطا یک چالش اساسی است که این روش با فراهم آوردن سریع منابع جایگزین سعی در رفع این چالش کرده است و توجهی به استفاده درست از منابع نکرده است.

۷- رهیافت مبتنی بر کپی‌های پشتیبان [۱۰] (روش تکرار) روشی است که در آن برای انجام یک وظیفه چندین کپی از آن وظیفه بر روی منابع مختلف ایجاد می‌شود: کپی اصلی و کپی‌های پشتیبان. بر اساس رهیافت مبتنی بر چندین کپی، کپی‌های پشتیبان به‌صورت موازی با کپی اصلی اجرا می‌شوند. در این روش به‌ازای هر وظیفه که وارد محیط ابر می‌شود، n تا کپی از آن وظیفه بر روی ماشین‌های مجازی دیگر ایجاد می‌شود. وظایف بر روی تمامی ماشین‌های مجازی اجرا می‌شوند. در صورتی که وظیفه بر روی ماشین مجازی اول دچار خطا شود خروجی بلافاصله به‌وسیله ماشین مجازی دوم تولید می‌شود و چنانچه وظیفه بر روی ماشین مجازی دوم نیز دچار خطا شود، خروجی به‌وسیله ماشین مجازی بعدی تولید می‌شود.

انتخاب مقدار n بزرگ‌ترین چالش در روش تکرار است زیرا به ازای هر وظیفه، nتا از این وظیفه بر روی n ماشین مجازی دیگر اجرا می‌شود. مقدار n با میزان قابلیت اطمینان ارتباط مستقیم دارد و با میزان استفاده بهینه از منابع رابطه معکوس دارد یعنی هر چه مقدار n بزرگ‌تر باشد قابلیت اطمینان سامانه بالاتر می شود و میزان بهینگی منابع بسیار پایین می شود و اگر مقدار n کمتر باشد میزان قابلیت اطمینان پایین‌تر و میزان بهینگی منابع بیشتر می‌شود [۱۲-۱۱].

با استفاده از این روش، احتمال از دست رفتن محدوده زمانی وظایف کاهش پیدا می‌کند ولی باعث افزایش مشغول بودن منابع محاسباتی رایانش ابری می‌شود.

در این روش، کپی های پشتیبان به همراه وظیفه اصلی بر روی منابع دیگر به طور همزمان اجرا می شود یعنی برای اجرای یک وظیفه با یک مقدار منابع، در این روش باید از چندین برابر آن منابع استفاده نماییم. و حتی اگر وظیفه اصلی ما دچار خطا نشود، کپی‌های پشتیبان اجرا می‌شود و باعث استفاده نادرست و غیر بهینه از منابع می‌شود.

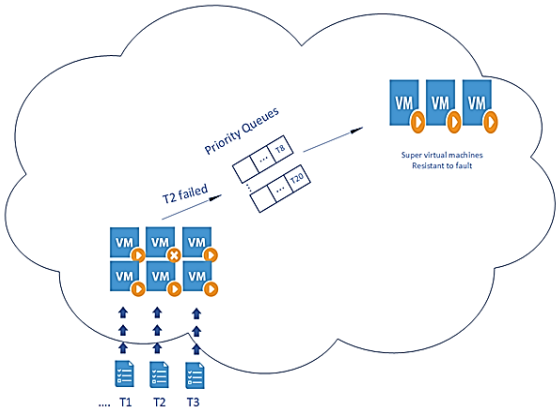
۲-۱- تحلیل کلی بر کارهای انجام شده

با بررسی کارهای مرتبط در زمینه تحمل‌پذیری خطا در رایانش ابری، مشاهده می‌شود که تمامی این روش‌ها سعی در تحمل‌پذیر کردن سرویس های ابری در مقابل خطاهای به‌وجود آمده برای

افزایش قابلیت اطمینان دارند. این کار از طریق زمان‌بندی مجدد وظایفی که دچار خطا شده‌اند انجام می‌گیرد. تمامی روش‌های ذکر شده، استفاده مناسب و بهینه از منابع را به صورت جدی مورد بررسی قرار نداده‌اند و بیشتر بر روی قابلیت اطمینان تمرکز کرده اند، در حالی که استفاده بهینه از منابع هم برای فراهم آورندگان سرویس‌های ابری و هم برای مشتریان سرویس‌های ابری بسیار مهم بوده و هر دو آن‌ها، سعی در استفاده از منابع با بازدهی بالا دارند. از این رو، نیاز است تا از طریق زمان‌بندی مناسب وظایفی که دچار خطا شده‌اند، یک روش تحمل‌پذیر خطا در محیط رایانش ابری ارائه داد که علاوه بر تحمل پذیر بودن در برابر خطاها، استفاده بهینه از منابع را نیز به‌صورت جدی مورد توجه قرار دهد.

۳- روش پیشنهادی

ایده اصلی روش پیشنهادی بر مبنای ایجاد خوشه‌ای از ماشین‌های مجازی است که این ماشین‌ها در برابر انواع خطاها مقاوم هستند. وظایفی که در حین اجرا دچار خرابی می‌شوند برای زمان‌بندی دوباره و اجرا وارد این خوشه از ماشین‌های مجازی می‌شوند. زمان‌بندی وظایف خراب‌شده برای ورود به این خوشه از طریق صف‌های اولویت‌دار صورت می‌گیرد. در شکل ۲، معماری روش پیشنهادی را مشاهده می‌کنید. در ادامه به تشریح قسمت‌های مختلف این معماری پرداخته می‌شود.



شکل (۲): معماری روش پیشنهادی.

در محیط رایانش ابر، ماشین‌های مجازی مسئول پردازش وظایف هستند. در روش پیشنهاد داده شده، دو نوع ماشین مجازی به نام‌های ماشین‌های مجازی معمولی و ابر ماشین‌های مجازی وجود دارند.

ماشین‌های مجازی معمولی سازوکاری برای مقاوم شدن در برابر خطاها ندارد و تنها وظیفه اجرای وظایف ارسال شده را دارند ولی در ابر ماشین‌های مجازی با استفاده از افزونگی‌های سخت‌افزاری و نرم‌افزاری، سعی شده است تا این ماشین‌ها در برابر خطاها مقاوم باشند. لازم به ذکر است که دست‌یابی به تحمل‌پذیری خطای صد در صد، امکان‌پذیر نمی‌باشد، زیرا ممکن است در هنگام اجرای وظایف، شرایطی به‌وجود آید که از قبل دانشی نسبت به آن وجود نداشته است. بنابراین، همیشه سعی می‌شود تا درصد تحمل‌پذیری خطا به بالاترین میزان برسد. استفاده از افزونگی باعث استفاده بیشتر از منابع می‌شود ولی ما در روش ارائه شده، از تعدادی بسیار محدود از ابر ماشین‌ها برای تحمل‌پذیری خطا استفاده می‌کنیم. انتخاب تعداد ماشین‌های مجازی برای قرار گرفتن در این خوشه، با توجه به اندازه و بزرگی محیط رایانش ابری صورت می‌گیرد. هم‌چنین تعداد ابر ماشین‌های مجازی، رابطه مستقیم با میزان منابع مصرفی دارد یعنی هر چقدر از تعداد ابر ماشین‌های مجازی بیشتری استفاده نماییم میزان استفاده از منابع مصرفی بیشتر می‌شود. بنابراین، باید تعداد ابر ماشین‌های مجازی را به نحوی انتخاب کنیم که باعث استفاده بی‌رویه از منابع نشود و همچنین باعث افزایش بیش از حد زمان پاسخ نیز نشود.

وظایف از سوی مشتریان برای اجرا وارد ماشین‌های مجازی معمولی می‌شوند. ماشین‌های مجازی معمولی وظایف ورودی را پردازش می‌کنند و پس از اجرای کامل و بدون خطای وظایف، خروجی موردنظر را تولید می‌کنند. ولی در حین پردازش وظایف، به دلایل مختلف احتمال بروز خطا وجود دارد. در صورت بروز خطا در وظایف، آن وظایف برای تولید خروجی باید دوباره اجرا شوند. در روش پیشنهادی، وظایفی که دچار خطا شده‌اند برای اجرای دوباره به صف‌های اولویت‌دار ارسال می‌شوند. در صف‌های اولویت دار با توجه به پارامترهای هر وظیفه، به هر یک از وظایف ورودی، یک اولویت اختصاص داده می‌شود. پس از اختصاص اولویت، وظایف بر اساس اولویت‌شان برای اجرای دوباره، به خوشه ابر ماشین‌های مجازی ارسال می‌گردند. این ابر ماشین‌ها با استفاده از روش‌های افزونگی در برابر خطاها مقاوم می‌باشند. تعداد ابر ماشین‌های مجازی بسیار کمتر از ماشین‌های مجازی عادی است. به همین دلیل میزان منابع و انرژی مصرفی در روش پیشنهادی بسیار کمتر از روش‌های معمول تحمل‌پذیری خطا است.

۳-۱- الگوریتم زمانبندی وظایف خراب شده

وظایفی که دچار خطا شده‌اند برای زمان‌بندی دوباره وارد صف‌های اولویت‌دار می‌شوند. در این صف‌ها برای هر کدام از این وظایف یک اولویت اختصاص داده می‌شود تا براساس اولویت‌شان وارد ابر ماشین‌های مجازی شوند. تعیین اولویت وظایف بسیار مهم است؛ زیرا اولویت وظایف، ترتیب اجرای وظایف را مشخص

می‌کند و ترتیب اجرای وظایف تأثیر مستقیم بر زمان اجرای کل (زمان پاسخ) دارد. بنابراین، ما باید ترتیبی از وظایف را برای اجرا در ابر ماشین‌های مجازی در نظر بگیریم که زمان پاسخ تا حد ممکن کمتر شود.

برای این‌که زمان‌بندی طراحی کنیم که تا حد ممکن زمان پاسخ را بهینه کند نیاز است تا ابتدا با نوع وظایف موجود در محیط ابری آشنا شویم.

معمولاً در محیط ابر به‌منظور افزایش سرعت اجرای کارها و وظایف، وظایف به تعدادی زیر وظیفه تقسیم می‌شوند و در قالب‌های مختلفی مانند معماری ارباب/برده اجرا می‌شوند. در این معماری یک گره، وظیفه تقسیم‌کارها به زیر وظایف را بر عهده دارد و تعدادی گره دیگر، وظیفه اجرای این زیر وظیفه‌ها را بر عهده دارند. در این محیط در بسیاری از مواقع بین زیر وظایف نوعی وابستگی وجود دارد. به‌عنوان مثال یک زیر وظیفه برای این‌که بتواند قابلیت اجرایی به‌دست آورد نیازمند خروجی‌های تولید شده توسط چندین زیر وظیفه دیگر است. در نظر گرفتن این وابستگی بین زیر وظایف بسیار مهم است زیرا این وابستگی در زمان‌بندی وظایف تأثیر دارد و برای این‌که بتوانیم زمان‌بندی مناسبی را طراحی کنیم باید توجه ویژه‌ای به وابستگی بین زیر وظایف داشته باشیم.

با توجه به مطالب بیان شده برای رسیدن به زمان پاسخ مناسب می‌بایست وظیفه‌ای که در زمان کمتر انجام می‌گیرد و تعداد وظایف منتظر بیشتری دارد و همچنین مجموع زمان‌های منتظر این وظیفه کمتر است، زودتر اجرا شود. یعنی در صف اولویت‌دار، اولویت بالاتری را به این وظایف اختصاص بدهیم. با توجه به مطالب گفته شده، پارامترهای ۱ و ۲ و ۳ دخیل در تخصیص اولویت به وظیفه هستند:

(۱) $T=$ زمان اجرای وظیفه جاری

(۲) $S(T) =$ تعداد وظایف منتظر وظیفه جاری

(۳) $T_{S(T)} =$ مجموع زمان‌های وظایف منتظر وظیفه جاری

همان‌طور که بیان کردیم برای رسیدن به زمان پاسخ بهتر، وظیفه‌ای که دارای زمان کمتر و تعداد وظیفه منتظر بیشتر و هم‌چنین مجموع زمان‌های منتظر این وظیفه کمتر است باید زودتر اجرا شود. از آن‌جایی که این متغیرها از یک جنس نیستند و ما هیچ بینشی نسبت به میزان تأثیر هر کدام از آن‌ها نداریم، لذا نمی‌توان مقایسه معقول و منطقی را برای ترکیب این متغیرها در نظر گرفت. درنتیجه برای این‌که بتوان سناریوی مناسبی را برای ترکیب این متغیرها درنظر گرفت، می‌بایست از نسبت زمان به تعداد کار برای هر وظیفه استفاده نماییم و هر چه این نسبت برای یک وظیفه مقدار کمتری باشد، وظیفه مناسبی است چون در زمان کمتر انجام می‌گیرد و تعداد کارهای بیشتری با انجام این وظیفه، قابلیت اجرایی پیدا می‌کنند. بنابراین، با توجه به توضیحات داده شده رابطه‌های ۱، ۲، ۳ و ۴ به‌دست می‌آیند:

(۱) $P(T) =$ اولویت وظیفه جاری

(۲) $P(T) \alpha T_{S(T)}$

(۳) $P(T) \alpha T$

(۴) $P(T) \alpha 1/S(T)$

با توجه به این‌که خود وظیفه جاری نیز یک وظیفه می‌باشد، بنابراین، رابطه (۵) برای تخصیص اولویت به وظایف، مورد استفاده قرار می‌گیرد.

(۵) $P(T) =$ اولویت وظیفه جاری $= (T_{S(T)}+T)/(S(T)+1$

هر وظیفه‌ای که اولویت آن بر اساس فرمول بالا، پایین‌تر باشد به‌عنوان وظیفه مناسب انتخاب می‌شود و برای اجرا به ابر ماشین مجازی فرستاده می‌شود.

۴- ارزیابی نتایج

در این قسمت با استفاده از نرم‌افزار شبیه‌سازی کلودسیم، معماری پیشنهادی را شبیه‌سازی می‌کنیم و نتایج حاصل از روش ارائه شده را با نتایج حاصل از شبیه‌سازی روش تکرار که نسبت به روش‌های دیگر تحمل پذیری خطا از میزان بهینگی منابع بیشتری پشتیبانی می‌کند و اساس و پایه روش‌های دیگر تحمل‌پذیری خطا است مقایسه نموده و میزان بهبود در روش ارائه‌شده را نشان می‌دهیم.

۴-۱- پارامترهای اولیه

پارامترهای اولیه که برای شبیه‌سازی هر دو روش (روش پیشنهادی و روش تکرار) استفاده شده است، در جدول (۱) نشان داده شده است.

جدول (۱): پارامترهای اولیه شبیه‌سازی.

Task Numbers	۲۰۰
Task Length	به‌صورت تصادفی بین ۱۰۰و۲۰۰۰
Data Center Number	۱
Host Numbers	۱

هم‌چنین پیکربندی نمونه ماشین مجازی استفاده شده مطابق جدول (۲) می‌باشد.

جدول (۲): پیکربندی نمونه ماشین مجازی.

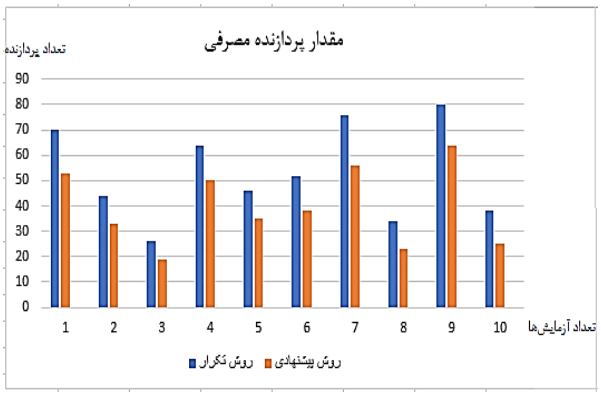
حافظه مصرفی	ناظر ماشین‌مجازی	پهنای‌باند مصرفی	توان پردازشی هر هسته	تعداد هسته پردازنده مصرفی
۵۱۲ مگابایت	ای اس ایکس آی	۱۰۰۰مگابیت بر ثانیه	۱۰گیگاهرتز	۱ هسته

با توجه به اینکه در روش پیشنهادی از ابرماشین‌های مجازی نیز استفاده می‌کنیم، پیکربندی زیر را برای ابرماشین‌های مجازی به منظور تحمل‌پذیر بودن در برابر خطاها در نظر گرفته‌ایم.

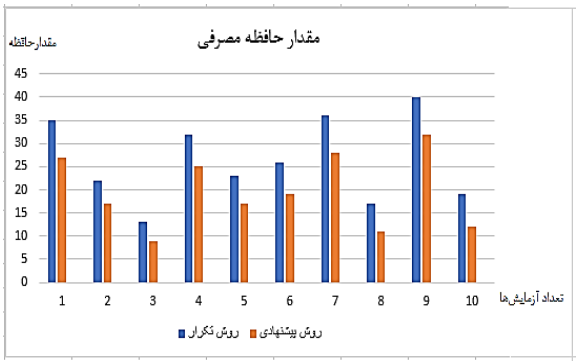
جدول (۳): پیکربندی نمونه ابر ماشین مجازی.

حافظه مصرفی	ناظر ماشین‌مجازی	پهنای‌باند مصرفی	توان پردازشی هر هسته	تعداد هسته پردازنده مصرفی
۳۰۷۰ مگابایت	ای اس ایکس آی	۶۰۰۰ مگابیت بر ثانیه	۱۰گیگاهرتز	۶ هسته

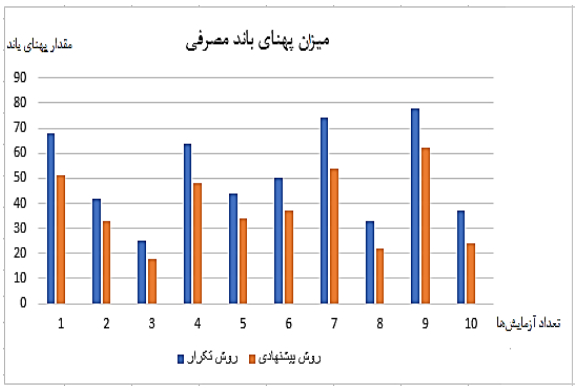
به منظور مقایسه دو روش ما ۱۰ آزمایش مختلف با تعداد ماشین های مجازی متفاوت در هر بار آزمایش، انجام دادیم که شکل‌های (۵-۳) میزان استفاده از منابع را در هردو روش نشان می‌دهد.



شکل (۳): تعداد پردازنده مصرفی در هر دو روش.



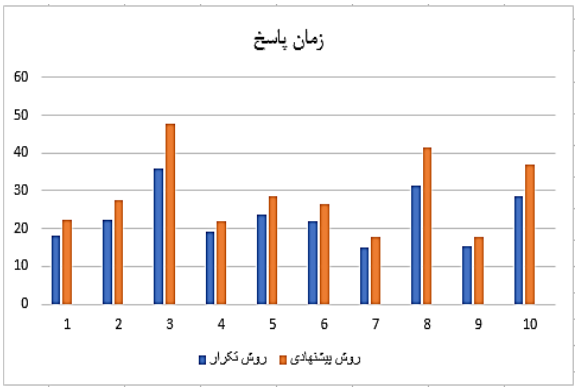
شکل (۴): مقدار حافظه مصرفی در هر دو روش.



شکل ۵: مقدار پهنای باند مصرفی در هر دو روش.

همان‌طور که از شکل‌های (۵-۳) مشخص است روش پیشنهادی توانسته است میزان استفاده از منابع (پردازنده، حافظه و پهنای باند) را، در مقایسه با روش تکرار کاهش دهد.

هم‌چنین شکل (۶) زمان پاسخ هر دو روش را در ۱۰ بار آزمایش نشان می‌دهد.



شکل (۶): زمان پاسخ هر دو روش.

با توجه به شکل (۶) مشاهده می‌گردد که روش تکرار در مقایسه با روش پیشنهادی در آزمایش‌های مختلف دارای زمان پاسخ کمتری می‌باشد اما با توجه به این نکته که ما در روش ارائه شده به کاربردهای غیر بی‌درنگ، تمرکز کرده‌ایم بنابراین، زمان پاسخ، خیلی اهمیت ندارد و استفاده بهینه از منابع مورد توجه قرار گرفته است.

۴-۲- تطابق و راستی آزمایی نتایج روش پیشنهادی

به منظور مقایسه نتایج به‌دست‌آمده و تطابق و درستی نتایج حاصل از مقایسه، نسبت به محاسبه فاصله اطمینان میزان توان پردازشی مصرف شده توسط روش پیشنهادی با کارمشابه ارائه شده اقدام شد. جدول (۴)، رابطه‌های لازم جهت محاسبه فاصله اطمینان را نشان می‌دهد.

جدول (۴): رابطه‌های مربوط به محاسبه فاصله اطمینان

$Sx = \sqrt{\frac{\sum(x-\bar{x})^2}{n-1}}$ = انحراف معیار

x = پیشامد مشاهده شده

$\bar{x}$ = میانگین پیشامدهای مشاهده شده

n = تعداد پیشامد مشاهده شده

$\left(\bar{x} - t_{\frac{\alpha}{2}}\left(\frac{S_x}{\sqrt{n}}\right), \bar{x} + t_{\frac{\alpha}{2}}\left(\frac{S_x}{\sqrt{n}}\right)\right)$ = اطمینان بازه

t= توزیع تی استونت

فاصله اطمینان تخمینی از دامنه‌ای را که میانگین و یا دیگر مقیاس‌های اندازه گیری، مثل درصد مشاهدات مربوط به یک جامعه خاص که باید در آن قرار گیرد، فراهم می آورد. این دامنه مبتنی بر پاسخ‌های جمع‌آوری شده از نمونه، تعداد مشاهدات و درجه خطای پذیرفته شده است.

دامنه میانگین که با فاصله اطمینان پیش‌بینی می‌شود، دلالت دارد بر این‌که شما قطعا دارای درصد احتمال هستید تا میانگین جامعه پاسخ دهندگان در محدوده قابل قبولی از میانگین نمونه قرار گیرد. همچنین فاصله‌ای اطمینان را می‌توان برای درصدی از پاسخ‌دهندگان محاسبه کرد.

رابطه‌های لازم برای برآورد فاصله اطمینان در جدول (۴) به نمایش درآمده است. سطح اطمینان، خطای معیار و ضریب اعتماد توزیع آماری دارای رابطه مستقیم و حجم نمونه‌ای دارای رابطه معکوس با فاصله اطمینان برآورده شده است.

در این پژوهش فاصله اطمینان با سطح اطمینان ۹۵ درصد برآورد شده است. با توجه به ماهیت متغیر (توان پردازشی) و آزمایش‌های انجام گرفته، توزیع احتمالی که به منظور تعیین فاصله اطمینان به‌کارگرفته شده است یک توزیع شبه نرمال به نام توزیع تی‌استونت است. حجم نمونه‌ای ما در این پژوهش برابر ۱۰، معادل با تعداد آزمایش‌های انجام شده است.

با توجه به فرمول‌های جدول (۴) روش پیشنهادی با ضریب اطمینان ۹۵ درصد، به ۲۹/۱۵٪ و ۲۲/۷۴٪ بهبود در استفاده از تعداد پردازنده نسبت به روش تکرار دست یافته است. هم‌چنین روش پیشنهادی با ضریب اطمینان ۹۵ درصد، به ترتیب به ۳۰/۷۶٪ و ۲۲/۳۴٪ و ۲۹/۷۱٪ و ۲۲/۸۸٪ بهبود در استفاده از مقدار حافظه و پهنای باند دست یافته است.

۵- نتیجه‌گیری

در سال‌های اخیر رایانش ابری در حال تبدیل شدن به یک فناوری مهم در حوزه فناوری اطلاعات است. در محیط رایانش ابری مانند تمامی سامانه‌های دیگر احتمال بروز خطا وجود دارد. روش‌های متفاوتی برای مقابله با خطاها وجود دارد ولی با توجه به ویژگی‌ها و خصوصیت‌های محیط رایانش ابری، استفاده از روش‌های تحمل‌پذیری خطا، بهترین انتخاب برای این محیط می‌باشد. روش‌های متفاوتی برای تحمل‌پذیر خطا در محیط رایانش ابری وجود دارد که هر کدام از این روش‌ها سعی در بهبود عملکرد و کارایی محیط رایانش ابری در برابر خطاها دارند. یکی از بزرگ‌ترین چالش‌ها در روش‌های تحمل‌پذیری خطا استفاده بهینه از منابع می‌شود. استفاده بهینه از منابع هم برای فرآهم آورندگان سرویس‌های ابری و هم برای مشتریان سرویس‌های ابری دارای اهمیت زیادی است. ما در این مقاله یک روش تحمل‌پذیری خطا را ارائه دادیم که علاوه بر تحمل‌پذیر بودن در برابر خطا، بهینگی در استفاده از منابع را در نظر گرفته است. در روش پیشنهادی ما، خوشه‌ای از منابع با تعداد محدود اما با افزونگی بالا برای تحمل‌پذیر بودن در برابر خطاها در نظر گرفته‌شده‌اند. وظایف در ابتدا به ماشین‌های مجازی معمولی فرستاده می‌شوند اگر وظیفه‌ای در هنگام اجرا دچار خطا شود برای اجرای دوباره به ابر ماشین‌های مجازی فرستاده می‌شوند. وظایف قبل از ارسال به ابر ماشین‌های مجازی در صف‌های اولویت‌دار قرار می‌گیرند. در این صف به هر یک از وظایف براساس پارامترهای هر وظیفه یک اولویت اختصاص داده می‌شود و وظایف بر اساس اولویت‌شان به ابر ماشین‌های مجازی فرستاده می‌شوند. زمان‌بندی وظایف براساس اولویت‌شان جهت دست‌یابی به زمان پاسخ بهتر استفاده شده است. استفاده بهینه از منابع در تحمل‌پذیر کردن محیط رایانش ابری در برابر خطاها مهم‌ترین دست آورد این مقاله می‌باشد.

باتوجه به این‌که قابلیت اطمینان و دسترس‌پذیری، دو ویژگی بسیار مهم برای سامانه‌های ابری می‌باشند، به‌عنوان کارهای آتی ما بر روی افزایش قابلیت اطمینان و قابلیت دسترس‌پذیری روش پیشنهادی تمرکز می‌کنیم. در ابتدا میزان اطمینان و دسترس پذیری روش پیشنهادی را محاسبه می‌کنیم و سپس سعی می کنیم تا روشی جهت بهبود این پارامترها در روش پیشنهادی ارائه دهیم. همچنین با توجه به اهمیت سرویس‌های بی‌درنگ در محیط محاسبات ابری در نظر داریم تا با بهبود زمان پاسخ روش پیشنهادی، بتوان از این روش برای سرویس‌های بی‌درنگ نیز استفاده کرد.

۶- مراجع

[1] H. Kaur and A. Kaur, "A Survey on Fault Tolerance Techniques in Cloud Computing Envirenment," International Journal of Scientific Engineering and Applied Science, vol. 1, 2015.

[2] M. Cheraghlou, A. Khadem-Zadeh, and M. Haghparast, "A Survey of Fault Tolerance Architecture in Cloud Computing," Journal of Network and Computer Applications, pp. 81-92, 2015.

[3] S. Prathiba and S. Sowvarnica, "Survey of Failures and Fault Tolerance in Cloud," 2nd International Conference on Computing and Communications Technologies (ICCCT), 2017.

[4] K. Plankensteiner et al, "Fault-Tolerant Behavior in State of the Art Grid Workflow Management Systems," CoreGRID Integration Workshop, Integrated Research in Grid Computing, 2008.

[5] Y. Liu and W. Wei, "A Replication-Based Mechanism for Fault Tolerance in Mapreduce Framework," Mathematical Problems in Engineering, pp 1-7, 2015.

[6] S. Malik and F. Huet, "Adaptive Fault Tolerance in Real Time Cloud Computing," IEEE World Congress on Services, 2011.

[7] A. D. Meshram, A. Sambare, and S. Zade, "Fault Tolerance Model for Reliable Cloud Computing", International Journal on Recent and Innovation Trends in Computing and Communication, vol. 1 pp. 600-603, 2013.

[8] X. Zhu et al, "Fault-Tolerant Scheduling for Real-Time Scientific Workflows With Elastic Resource Provisioning in Virtualized Clouds," in IEEE Transactions on Parallel and Distributed Systems, 2016.

[9] X. Zhu, X. Qin, and M. Qiu, "QoS-aware Fault-Tolerant Scheduling for Real-Time Tasks on Heterogeneous Clusters," in IEEE Transactions on Computers, 2011.

[10] T. Altameem, "A Replication-Based and Fault Tolerant Allocation Algorithm for Cloud Computing," International Journal of Computer Science Engineering and Technology, vol. 4, pp. 395-399, 2014.

[11] R. Al-Omari, A. K. Somani, and G. Manimaran, "Efficient Overloading Techniques for Primary-Backup Scheduling in Real-Time Systems," Journal of Parallel and Distributed Computing, vol. 64, pp. 629-648, 2004.

[12] S. Ghosh, R. Melhem, and D. Mossé, "Fault-tolerance through scheduling of aperiodic tasks in hard real-time multiprocessor systems," in IEEE Transactions on Parallel and Distributed Systems, 1997.

Journal of Electronical & Cyber Defence
Vol. 8, No. 2, 2020, Serial No. 30

A Fault Tolerant Task Scheduling Method for Optimal use of Resources in Cloud Computing Environment

M. J. Ahrari, M. R. Hasani Ahangar*, A. Ghafouri

*Imam Hosein University

(Received: 05/08/2019, Accepted: 28/06/2019)

ABSTRACT

In recent years, cloud computing is becoming eminent in the field of information technology. In a cloud computing environment, there is a potential for faults. There are different methods for dealing with faults, but with regard to the features and characteristics of the cloud computing environment, the use of fault tolerance methods is the best choice for this environment.One of the biggest issues in fault tolerance methods is the efficient use of resources. The optimal use of resources is important for cloud providers and customers. Unfortunately, the optimal use of resources in fault tolerance methods has not been much considered by researchers and cloud service providers. In this paper taking into account the dependence between tasks, an attempt has been made to provide a fault tolerance method on virtual machines, which in addition to being tolerant of fault, achieves optimum use of resources. In this method, by using a priority scheduler, each task is assigned a priority, then tasks are sent by the order of priority to their virtual machines for processing. The results of simulation by the cloudsim simulator show that the proposed method has been able to improve the use of resources more than other methods and with 95% confidence intervals it has achieved (29.15% and 22.74%) improvement in the number of processors, (30.76% and 22.34%) improvement in memory usage and (29.71% and 22.88%) improvement in the use of bandwidth.

Keywords: Cloud Computing, Fault Tolerance, Virtual Machines, Optimization of Resources, Task Scheduling

* Corresponding Author Email: mrhasani@ihu.ac.ir