

مدلسازی و ارزیابی تحمل پذیری خطا و قابلیت اطمینان در محیط ابری با استفاده از شبکه های پتری رنگی

سعد بهمنی^۱، بهنام برزگر^{۲*}

^۱ کارشناسی ارشد فناوری اطلاعات، دانشکده فنی مهندسی، موسسه آموزش عالی روزبهان، ساری، ایران.
^۲ استادیار گروه مهندسی کامپیوتر، دانشکده فنی مهندسی، دانشگاه آزاد اسلامی واحد بابل، بابل، ایران.

چکیده

امروزه، ارائه خدمات قابل اعتماد به مشتریان به عنوان موضوعی مهم برای ارائه دهندگان خدمات تبدیل شده است. طراحی سیستم هایی با پیچیدگی بالا و وجود منابع مختلف در شبکه ابر باعث شده تا ارائه دهندگان خدمات، قصد ارائه بهترین سرویس ها را به مشتریان خود داشته باشند. یکی از چالش های مهم فراهم کنندگان سرویس های ابری، افزایش تحمل پذیری در برابر خطا و نیز دستیابی به حداکثر قابلیت اطمینان است، به طوری که روش های مختلفی برای حل این چالش ارائه شده است. در این مقاله، فرایند تحمل پذیری خطا را در ابر به منظور جلوگیری از بروز مشکلات و خسارات جبران ناپذیر قبل از اجرا، تجزیه و تحلیل می نماییم. در روش پیشنهادی، تحمل پذیری خطا در محیط ابری با استفاده از شبکه های پتری رنگی، مدلسازی و ارزیابی شده است. روش پیشنهادی را با استفاده از نرم افزار CPN Tools مدلسازی کرده و پارامتر قابلیت اطمینان مورد ارزیابی قرار گرفته است. نتایج مدلسازی و شبیه سازی نشان می دهد که با افزایش تعداد درخواست ها، تحمل پذیری خطا و در نتیجه قابلیت اطمینان کاهش می یابد.

کلمات کلیدی: تحمل پذیری خطا، قابلیت اطمینان، شبکه های پتری رنگی، رایانش ابری.

Modeling and Evaluating Fault Tolerance and Reliability in Cloud Environment using Colored Petri Nets

Saed Bahmani¹, Behnam Barzegar^{2*}

¹ Master of Information Technology, Faculty of Engineering, University College of Rouzbahan, Sari, Iran.

² Assistant Professor, Department of Computer Engineering, Faculty of Engineering, Islamic Azad University of Babol Branch, Babol, Iran.

Abstract

Nowadays, providing reliable customer service has become an important issue for service providers. The design of systems with high complexity and the existence of different resources in the cloud network have made service providers intend to provide the best services to their customers. One of the important challenges for cloud service providers is to increase fault tolerance as well as achieve maximum reliability, so that different methods have been proposed to solve this challenge. In this paper, we analyze the fault tolerance process in the cloud to prevent problems and irreparable damage before implementation. In the proposed method, the fault tolerance in the cloud environment is modeled and evaluated using colored petri nets. The proposed method is modeled using CPN Tools software and the reliability parameter is evaluated. Modeling and simulation results show that with increasing number of requests, Fault tolerance and reliability are reduced.

تاریخچه مقاله:

تاریخ ارسال:	۱۳۹۹/۱۰/۱۱
تاریخ اصلاحات:	۱۳۹۹/۱۲/۲۰
تاریخ پذیرش:	۱۳۹۹/۱۲/۲۵
تاریخ انتشار:	۱۳۹۹/۱۲/۲۹

Keywords:

Fault Tolerance
Reliability
Colored petri nets
Cloud Computing

* ایمیل نویسنده مسئول:

Barzegar.Behnam@yahoo.com

کم شوند. تکنیک های تحمل پذیری خطا برای پیش بینی کردن این خطاها استفاده می شوند و یک عملی را قبل یا بعد از اینکه خطا رخ داده است اختصاص می دهند [۲ و ۳].

شبکه های پتری سیستم های پویایی هستند که با مفاهیمی همچون همروندی، همگام سازی و انحصار متقابل سروکار دارند و عموماً بصورت محیط های توزیع شده هستند. از شبکه های پتری می توان برای نمایش ارتباطات علت و معلول استفاده نمود. شبکه های پتری یک ابزار توصیف بصری و مبتنی بر اصول ریاضی است که به صورت یک گراف متصل و جهت دار نشان داده می شود. معمولاً، شبکه های پتری را از دو دیدگاه ساختاری و رفتاری مورد بررسی قرار می دهند. در دیدگاه ساختاری، اجزا و ساختار یک شبکه پتری و مفاهیم مقدماتی مورد بحث قرار می گیرند. در دیدگاه رفتاری، نحوه نشانه گذاری و اجرای شبکه های پتری توسط نشانه ها و قواعد اجرا و همچنین ویژگی های رفتاری شبکه های پتری مورد بررسی قرار می گیرد [۴ و ۵].

یکی از شبکه هایی که بر اساس شبکه پتری کلاسیک ارائه شده است، شبکه پتری رنگی می باشد. در این شبکه علاوه بر گذر، خال و مکان، از مفاهیم گارد، رنگ و عبارات استفاده می شود. در شبکه های پتری رنگی مقادیر داده ای با خال ها جا به جا خواهند شد. برای مدلسازی سیستم های پردازشی پیچیده و غیرهمزمان از شبکه های پتری رنگی استفاده می شود که برخلاف شبکه های پتری کلاسیک، خال ها با داشتن مقدار رنگ از یکدیگر جدا می شوند [۶].

هدف از این پژوهش، مدلسازی و ارزیابی تحمل پذیری خطا و قابلیت اطمینان در محاسبات ابری با استفاده از شبکه های پتری رنگی می باشد. در این تحقیق تحمل پذیری خطا با استفاده از شبکه های پتری رنگی ارزیابی می شود.

۲ - پیشینه تحقیق

محاسبات ابری نوعی محاسبات است که در آن سرویس با استفاده از مدل ها و لایه های مختلف انتزاع در سراسر اینترنت ارائه می شود. این برنامه به برنامه های ارائه شده به عنوان خدمات انبوه اشاره دارد، از کاربران نهایی که اسناد شخصی خود را در اینترنت میزبانی می کنند تا شرکت هایی که کل زیرساخت فناوری اطلاعات^۱ خود را به مراکز داده خارجی برون سپاری می کنند. این شبکه ها به عنوان یک الگوی جدید محاسباتی ظاهر می شود که هدف آن فراهم آوردن محیط های پویا محاسباتی قابل اطمینان، سفارشی و تضمین شده با کیفیت

رایانش ابری، امکان دسترسی به محدوده وسیعی از منابع مجازی را فراهم می آورد که این منابع می توانند به صورت پویا و با حجم متغیر به کاربران خود سرویس دهند، یعنی هر کاربر به اندازه های که منابع نیاز دارد، از آن منابع استفاده می کند که این کار به استفاده بهینه از منابع منجر می شود. رایانش ابری، باید به صورت خودکار منابع مورد نیاز را به کاربران تخصیص دهد و اگر کاربری مقداری بیشتر از منابع را خواستار باشد، باید آن منابع بدون آنکه سرویس کاربر با مشکل مواجه شود، به مشتری تخصیص یابد و در صورتیکه منابع مورد استفاده بیش از حد مورد نیاز باشد، باید منابع اضافی به صورت موقت خاموش شده تا در صورت نیاز مجدداً مورد استفاده قرار گیرند [۱].

محاسبات ابری یک الگوی محاسبات توزیع شده در حال ظهور است که امروزه به یکی از الگوهای محاسباتی بسیار محبوب تبدیل شده است. یکی از دلایل محبوبیت محاسبات ابری به دلیل خاصیت ارتجاعی آن است. در اصل محاسبات ابری به این معنا است که منبع یک نرم افزار، داده های آن و همینطور پردازش های مربوط به آن را به یک فضای بیرونی منتقل کنیم. کاربران می توانند به اپلیکیشن ها و داده ها با استفاده از ورود به این سیستم با استفاده از هر دستگاهی که به اینترنت متصل است دسترسی داشته باشند. اطلاعات و برنامه ها توسط یک فضای خارجی میزبانی می شوند و به جای یک هارد دیسک فیزیکی بر روی یک شبکه جهانی از مراکز داده های امن نگهداری می شوند. این کار باعث می شود تا قدرت پردازش افزایش پیدا کند، امکان اشتراک داده ها و همکاری بیشتر با سایر افراد راحت تر می شود و امکان دسترسی امن به این داده ها از طریق رایانه ها، تلفن همراه، تبلتها و ... را بدون توجه به موقعیت مکانی کاربر فراهم می کند. از آنجائیکه امروزه ضرورت ارائه خدمات قابل اعتماد به مشتریان برای ارائه دهندگان این خدمات بسیار بااهمیت می باشد، طراحی سیستم هایی با پیچیدگی بالا و وجود منابع مختلف در شبکه ابر باعث می شود که ارائه دهندگان خدمات قصد ارائه بهترین خدمات را به مشتریان خود داشته باشند. یکی از چالش های مهم ارائه دهندگان خدمات، تحمل پذیری خطا و قابلیت اطمینان است و تا به امروز روش های مختلفی برای حل این چالش ارائه شده است. تحمل پذیری خطا یک نگرانی بزرگ برای تضمین قابلیت دسترسی و قابلیت اطمینان از سرویس های بحرانی و نیز اجرای درخواست است. به منظور کم کردن فشردگی خطا در سیستم و اجرای درخواست، خطاها بایستی

¹IT

اجتناب کرد که باعث افزایش کارایی و قابلیت اطمینان و تحمل خطا در این سیستم ها گردید [۱۱].

پرتی کوماری^۵ و پارمیت کاور^۶ (۲۰۱۸) بر روی بررسی تحمل پذیری خطا در محاسبات ابری کار کردند که این پژوهش مروری جامع از مسائل مربوط به تحمل خطا در محاسبات ابری و تأکید بر مفاهیم مهم، جزئیات معماری و تکنیک ها و روش های پیشرفته ارائه می دهد. هدف این است که بینش هایی در مورد رویکردهای تحمل خطای موجود و همچنین چالش هایی که هنوز باید بر آنها غلبه کرد، ارائه داده شود. این نظرسنجی چند تکنیک امیدوار کننده را که ممکن است برای راه حل های کارآمد استفاده شوند برشمرده و همچنین جهت های مهم تحقیق در این زمینه را مشخص می کند [۱۲].

جینگ لی^۷ و همکاران (۲۰۱۵) پژوهشی تحت عنوان مدل سازی سرویس های صف بندی پیام با ضمانت اطمینان در محیط محاسبات ابری با استفاده از شبکه های رنگی پتری انجام دادند. یک رویکرد مدل سازی مبتنی بر دید^۸ برای مدل شبیه سازی با استفاده از شبکه های پتری رنگی^۹ ارائه داده شد. این مدل دارای ویژگی های مهم خدمات صف پیام در فضای ابری مانند تکرار، سازگاری پیام، مجازی سازی منابع و به ویژه ساز و کاری به نام مهلت دید است که در سرویس ها برای تضمین قابلیت اطمینان سیستم استفاده می شود. سرانجام، این مدل را از طریق آزمایش های مختلف تحت سناریوهای متنوع ارزیابی نموده تا از سنجش عملکردی مهمی مانند زمان تحویل کل پیام، تعداد انتظار و استفاده از مولفه ها استفاده کنیم. نتایج بینش قابل توجهی در مورد برنامه ریزی منابع و پیکربندی سیستم برای ارائه دهندگان خدمات برای تخمین و بهینه سازی عملکرد به دست می دهد [۱۳].

علی شهیدی نژاد (۱۳۹۹) موضوعی تحت عنوان مدیریت کشسانی در محاسبات ابری با استفاده از شبکه پتری رنگی کار کرد که در این تحقیق، مدلی برای بهبود خاصیت کشسانی با استفاده از شبکه پتری رنگی برای تأمین منابع در شبکه های ابری ارائه شد. نتایج عملکرد روش پیشنهادی تحت بار کار واقعی کشسانی، افزایش دقت و افزایش سرعت را در مقایسه با رویکردهای مشابه نشان میدهد [۱۴].

استاوروس سوراولاس^{۱۰} (۲۰۲۰) بر روی مدل سازی و شبیه سازی سیاست های تخصیص منابع در محاسبات ابری با استفاده از پتری نت

(کیفیت خدمات) برای کاربران نهایی است. به طور کلی، محاسبات ابری جنبه های جدیدی را در مدیریت منابع محاسباتی به ارمغان می آورد. منابع محاسباتی بی نهایت موجود در صورت تقاضا برای دیدگاه کاربران نهایی می باشد و تعهد در برابر کاربران ابر و استفاده کوتاه مدت از هر گونه محاسبات پیشرفته صفر است. اصل اساسی محاسبات ابری این است که داده های کاربر به صورت محلی ذخیره نمی شوند، بلکه در مرکز داده اینترنت ذخیره می شوند [۷ و ۸].

تحمل پذیری خطا یک نگرانی عمده برای تضمین در دسترس بودن و قابلیت اطمینان خدمات حیاتی و همچنین اجرای برنامه است. به منظور به حداقل رساندن تأثیر خرابی بر سیستم و اجرای برنامه، باید خرابی ها را پیش بینی و به طور پیشگیرانه برطرف کرد. تحمل پذیری خطا برای پیش بینی این خرابی ها و اقدام مناسب قبل از وقوع واقعی خرابی ها، از روشهایی استفاده می شود [۹]. آنجو بال^۲ و ایندرویر چانا^۳ (۲۰۱۲) تحقیقی بر روی تحمل پذیری خطا، چالش ها، تکنیک ها و پیاده سازی در محاسبات ابری انجام دادند که تکنیک های تحمل پذیری خطای موجود در محاسبات ابری بر اساس سیاست های آنها، ابزارهای استفاده شده و چالش های تحقیقاتی مورد بحث قرار گرفته است. ابر معماری سیستم مجازی ارائه شده است. در سیستم پیشنهادی تحمل پذیری خطای خودکار اجرا شده است. نتایج تجربی نشان می دهد که سیستم پیشنهادی می تواند با خطاهای نرم افزاری مختلف برای برنامه های سرور در یک محیط مجازی ابر مقابله کند [۱۰].

خدیجه متاجی خیری و همایون موتمنی (۱۳۹۴) پژوهشی تحت عنوان ارائه روشی جهت افزایش تحمل خطای سیستم تولید انعطاف پذیر^۴ با استفاده از شبکه های پتری رنگی انجام دادند که هدف در این مطالعه عیب یابی و تشخیص به موقع خطا و رفع آن قبل از شکست در خط تولید سیستم های تولید انعطاف پذیر با استفاده از مشخصات ماشین ها و نرخ خرابی و طول عمر آنها بوده است. برای این کار ابتدا با استفاده از شبکه های پتری تصادفی رنگی دستگاه ها در سیستم های تولید انعطاف پذیر را مدل سازی کرده و بعد از تشخیص خطا در مدل ایجاد شده خطا و خرابی آن با تعمیرات پیشگیرانه مرتفع گردید که این امر قابلیت بالای تجهیزات را حفظ کرده و از زمان بیکار تولید

⁵ PritiKumari
⁶ ParmeetKaur

⁷ Jing Li

⁸ VMA

⁹ CPN

¹⁰ Stavros Souravlas

² Anju Bala

³ Inderveer Chana

⁴ FMS

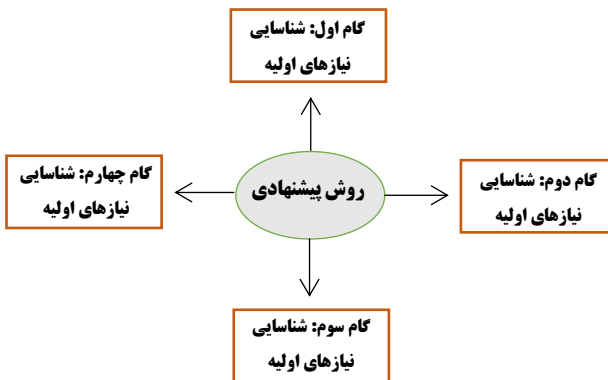
ساخت محاسبات ابری ارائه شده و سپس به تشریح همه مولفه های موجود در مدل پرداخته است. در ادامه به بیان روش پیشنهادی خواهیم پرداخت. ابتدا به مدل های تحقیق می پردازیم:

۱- مدل تئوری تحقیق: از جمله مشکلاتی که در زمینه تشخیص خطا وجود دارد، گستردگی انواع خطاها می باشد. در روش پیشنهادی با دسته بندی کردن خطاها در محاسبات ابری و بررسی تخصصی هر بخش به صورت مجزا، سعی در کاهش خطا دارد. روش پیشنهادی با در نظر گرفتن خطاها، می تواند بهترین ماشین مجازی برای اجرای درخواست کاربران را ارائه دهد. همچنین در نظر زمان اجرایی بسیار مناسب می باشد. در شکل ۱ این مدل نشان داده شده است.



(شکل ۱): مدل تئوری تحقیق

۲- مدل مفهومی تحقیق: هدف اصلی در این تحقیق، رسیدن به بیشترین قابلیت اطمینان در محاسبات ابری است. در نتیجه سعی بر این است تا با استفاده از الگوریتم پیشنهادی این میزان خطا را کاهش داده تا قابلیت خطا محاسبات ابری افزایش یابد. در شکل ۲ این مدل نشان داده شده است.



(شکل ۲): مدل مفهومی تحقیق

رنگی کار کردند که پتری نت به دلیل ویژگی های اساسی آن مانند همگام سازی، ابزاری مناسب برای مدل سازی سیستم های موازی است. در این پژوهش، چارچوب CPN برای پیاده سازی یک شبیه ساز تخصیص منابع جدید استفاده شده است، که برای تأیید عملکرد کارهای قبلی ما استفاده می شود، جایی که ما یک طرح تخصیص منابع عادلانه با کنترل جریان و حداکثر استفاده از منابع سیستم را ارائه می دهیم [۱۵].

جونائیک اوچو^{۱۱} و همکاران (۲۰۱۸) تحقیقی تحت عنوان بررسی اجمالی راهبردهای تحمل خطا در سرویس کامپوزیت در محاسبات سرویس انجام دادند که یک نمای کلی از استراتژی های اصلی تحمل خطا ارائه دادند. در مرحله اول، استراتژی های تحمل خطا با توجه به مرحله تصویب آنها، به تحمل خطای ایستا و پویا طبقه بندی می شوند. در مرحله دوم، استراتژی های مختلف تحمل خطای ایستا را بررسی شد. سپس، مکانیسم های اجرای تحمل گسل پویا مورد تجزیه و تحلیل قرار گرفت. سرانجام، چالشهای اصلی مقابله با تحمل خطا برای سرویس کامپوزیت بررسی شد [۱۶].

۳- راهکار پیشنهادی

یکی از اهداف مهم از اجرای طرح تحمل پذیری خطا در محاسبات ابری، افزایش قابلیت اطمینان در محیط ابر می باشد. تحمل پذیری خطا در محاسبات ابری مولفه ای بسیار مهم است، که آن را به عنوان تضمین کننده در دسترس بودن، قابلیت اطمینان و عملکرد برنامه های کاربردی در نظر خواهند گرفت. در برخی از مواقع، حجم کار زیاد و نوسانی محیط ابر به گونه ای است که ممکن است وضعیت اجرای برنامه ها به یک حالت نامعقول تبدیل شود. مولفه خطا برای دستیابی به قابلیت اطمینان، در دسترس بودن، عملکرد و استحکام محیط ابری امری ضروری است. حتی اگر خطاهای موجود در محاسبات ابری شناخته شده هم باشند، باز هم امکان بروز مشکلات وجود دارد.

تحمل پذیری خطا یکی از نگرانی های عمده برای تضمین در دسترس بودن و قابلیت اطمینان خدماتی حیاتی و هم چنین اجرای برنامه های کاربردی است. به منظور به حداقل رساندن تاثیر خرابی بر روی سیستم و اجرای صحیح و موفق برنامه های کاربردی، خرابی باید پیش بینی شده و فعالانه مدیریت و کنترل گردد. روش های تحمل خطا در واقع برای پیش بینی این خرابی ها و انجام یک اقدام مناسب قبل از خرابی می باشد. این بخش به بیان راهکار پیشنهادی پرداخته است. ابتدا مدلی جهت تحمل پذیری خطا و قابلیت اطمینان در زیر

¹¹ Junna Zhang Ao Zhou

۱- بافر ورودی:

مسئولیت وظایف ورودی به هر ابر را برعهده دارد.

۲- ماژول رم یا تخصیص منابع:

وظایف را از بافر ورودی دریافت کرده تا منابع مورد نیاز برای اجرا فراهم شود. پس از ورود وظایف به ابر، RAM بر اساس منابع مورد نیاز هر وظیفه، سرور مناسب را پیشنهاد می کند که در نهایت توسط مدیریت اصلی ابر، سرور مورد نظر انتخاب می شود.

۳- ماژول کنترل کننده ابرها:

کنترل اجرای وظایف و نتایج آنها بر عهده آن است. شرط اجرای وظایف این است که وظیفه از حد مشخص شده فراتر نرود.

۴- تست پذیرش:

هر سرور اجرا کننده در ابر دارای یک مولفه تست می باشد که تست پذیرش را برای خروجی هر وظیفه اعمال خواهد کرد و همچنین صحت و درستی نتیجه بدست آمده توسط وظیفه ورودی را روی سرور تایید می نماید. اگر پاسخ درست باشد، آنگاه آن وظیفه را به منظور بررسی و محاسبه قابلیت اطمینانش در قسمت وظایف درست به ارزیاب قابلیت اطمینان ارسال می کند، در غیر این صورت، برای بررسی و محاسبه قابلیت اطمینان منفی به قسمت ارزیاب قابلیت اطمینان منفی ارسال می گردد.

۵- ماژول ارزیابی قابلیت اطمینان:

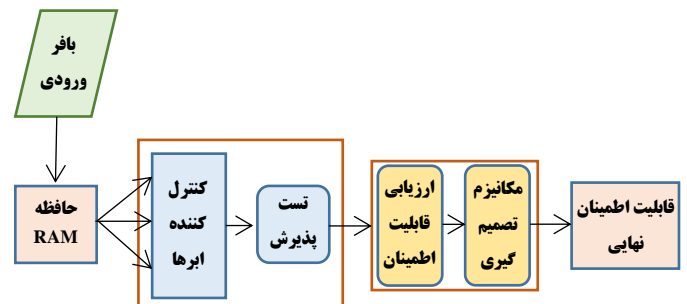
ماژول اصلی: محاسبه میزان قابلیت اطمینان برای هر وظیفه ورودی بر روی سرور است. الگوریتم پیشنهادی، خطاها و اشکالات را تحمل کرده و براساس قابلیت اطمینان وظیفه تصمیم گیری را انجام می دهد.

۶- ماژول مکانیزم تصمیم گیری:

خروجی ماژول ارزیاب قابلیت اطمینان می باشد. وظیفه انتخاب سرور را برعهده دارد. براساس قابلیت اطمینان سرور، در صورت اضافه شدن میزان قابلیت اطمینان از ماکزیمم مقدار در نظر گرفته شده یا کمتر شدن آن از مینیمم مقدار در نظر گرفته شده تصمیم گیری خواهد کرد. هر چه قابلیت اطمینان بالاتر باشد، به همان نسبت سرور انتخاب شده بهتر عمل می کند.

۱-۳- مدل پیشنهادی: جهت بهبود تحمل پذیری خطا

در محاسبات ابری ارائه گردیده و به منظور بهبود قابلیت اطمینان در پردازش ابری تحمل پذیری خطا مورد استفاده قرار می گیرد. تحمل پذیری خطا یک خصوصیت سیستمی است که اجازه میدهد به فراهم کردن سرویس درخواست شده که احتمالاً در معرض یک خطر وجود دارد در بالاترین سطح اطمینان اجرا شود. انواع مختلف خطاها ممکن است در ساختار ابر بروز پیدا کنند و همچنین انواع مختلف تکنیک های تحمل پذیری خطا در ابر استفاده می شود بر اساس کاربرد هایی که هر کدام از این تکنیک ها می توانند داشته باشند. این طرح، خطاها و اشکالات را بر اساس قابلیت اطمینان هر گره محاسباتی (ماشین مجازی) تحمل می کند. ماشین مجازی برای محاسبه براساس قابلیت اطمینانش انتخاب می شود. مسئله اصلی در مدل تحمل پذیری خطا آن است که کاملاً به کنترل کننده زمان تکیه می کند اما برای ایجاد مدل تحمل پذیری خطای درست، مجبور به در نظر گرفتن پارامترهای دیگر هستیم. در شکل ۳ مدل پیشنهادی قابلیت اطمینان نشان داده شده است.



(شکل ۳-): مدل پیشنهادی قابلیت اطمینان

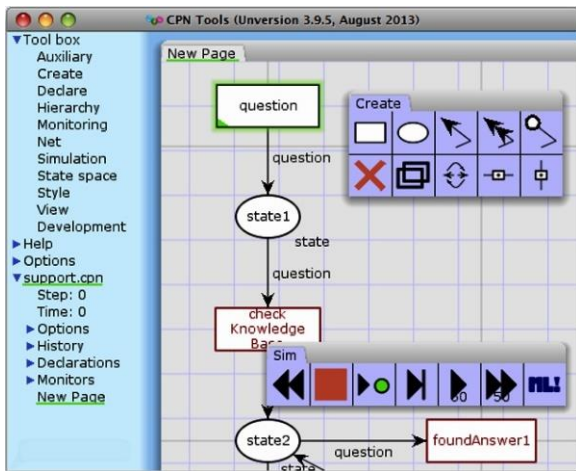
در مدل پیشنهادی، دو نوع داده در نظر گرفته شده است:

(۱) ماشینهای مجازی که روی زیرساخت ابری اجرا می شوند.

(۲) گره بازرس

کنترل کننده سطح اول حاوی کنترل کننده ابر است که روند انجام کار و تست پذیرش اعتبار را نظارت می کند. گره بازرس حاوی ارزیاب قابلیت اطمینان و مکانیزم تصمیم گیری است. این گره می تواند روی زیرساخت ابری و زیرساخت کاربر اجرا شود که این مسئله به نوع برنامه های کاربردی بستگی دارد.

شود. سیستم های اطلاعاتی فرایند محور همه وقایع رخ داده توسط سیستم و کاربر را به صورتی ساختاری در گزارش وقایعی ثبت می کنند و از روش های متعددی برای بیرون کشیدن دانش از اطلاعات واقع شده در این سیستم های اطلاعاتی استفاده می کند. CPN Tools یک ابزار کامپیوتری و شبیه سازی است که برای تجزیه و تحلیل مدل های شبکه ای مورد استفاده قرار می گیرد. این شبیه ساز می تواند تمامی مرحله چرخه حیات یک مسئله را نشان داده و تحلیل و ارزیابی کند. در شکل ۴ نمایی از شبیه سازی شبکه های پتری در این نرم افزار نمایش داده شده است.

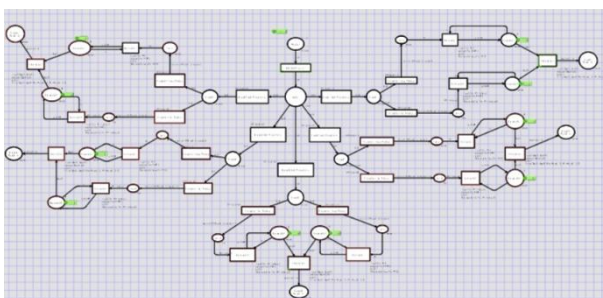


(شکل ۴): شمایی از پیاده سازی شبکه پتری در نرم افزار CPN Tools

تعدادی وظیفه به صورت تصادفی تعریف و مقاردهی می شود. هر وظیفه دارای ۲ مولفه می باشد که عبارتند از:

- شماره ابر اختصاص یافته
- رد یا پذیرش وظیفه

۳-۴- پیاده سازی: در این پژوهش از نرم افزار CPN Tools ورژن 4.0.1 ریلیز سال 2015 استفاده شده است. یک نمای کلی از شبکه پتری رنگی شبیه سازی شده در شکل ۵ نشان داده شده است.



(شکل ۵): شمایی کلی شبکه پتری پیاده سازی شده

۳-۲- مکانیزم تحمل پذیری خطا:

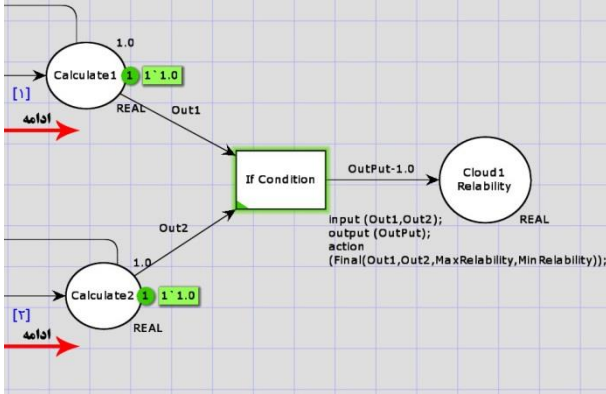
الگوریتم سنجش قابلیت اطمینان بصورت زیر می باشد:

1. **Begin**
2. Initially reliability:=1, n :=1
3. **Input** from configuration RF, maxReliability, minReliability
4. **Input** nodeStatus
5. **if** nodeStatus =Pass **then**
6. reliability:= reliability + (reliability * RF)
7. **if** n > 1 **then**
8. n:= n-1;
9. **else if** nodeStatus = Fail **then**
10. reliability := reliability – (reliability * RF * n)
11. n:= n+1;
12. **if** reliability >= maxReliability **then**
13. reliability := maxReliability
14. **if** reliability < minReliability **then**
15. nodeStatus:=dead
16. call_proc: remove_this_node
17. call_proc: add_new_node
18. **End**

۳-۳- شرایط و محیط شبیه سازی: الگوریتم پیشنهادی در نرم افزار CPN Tools شبیه سازی شده است و نتایج بدست آمده از آن در این مقاله بیان شده است.

نرم افزار CPN Tools یک شبیه ساز بسیار جامع و کامل بوده که در حوزه ارزیابی و کارایی الگوریتم های ارائه شده مورد استفاده قرار می گیرد. ارزیابی کارایی نرم افزار یکی از مسائل بسیار مهم در حوزه نرم افزار است، زیرا کارایی نرم افزار از جمله مولفه هایی است که باید پیش از پیاده سازی و اجرای نرم افزار مورد بررسی قرار گیرد. زبان مدل سازی یکپارچه یکی از بهترین و موثرترین مدل سازی ها برای ارزیابی معماری و کارایی سیستم های نرم افزاری است، اما نیمه رسمی بودن این زبان باعث شده است تا این زبان نتواند به ارزیابی تمامی نیازهای نرم افزاری بپردازد. شبکه های پتری یکی از روش های مدلسازی و شبیه سازی رسمی است که توانایی شبیه سازی انواع نیازهای نرم افزاری را دارا است. همچنین می تواند چرخه حیات نرم افزار را به صورت کامل و گام به گام پیاده سازی کند. در بسیاری از پژوهش هایی که در حوزه های مختلف انجام می شود نیاز به مجموعه داده ای از سیستم اطلاعاتی مورد ارزیابی برای مدل حس می

در شکل های ۶ و ۷ و ۸ و ۹ توکن ها و مراحل پیاده سازی را در محیط CPN Tools مشاهده می کنید.

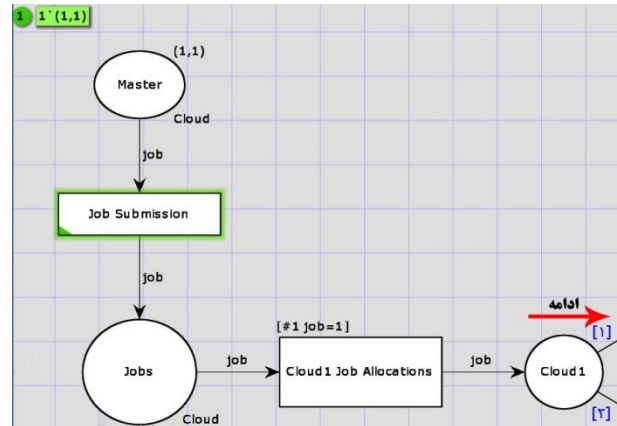


(شکل ۹): توکن ها و مراحل پیاده سازی ۴

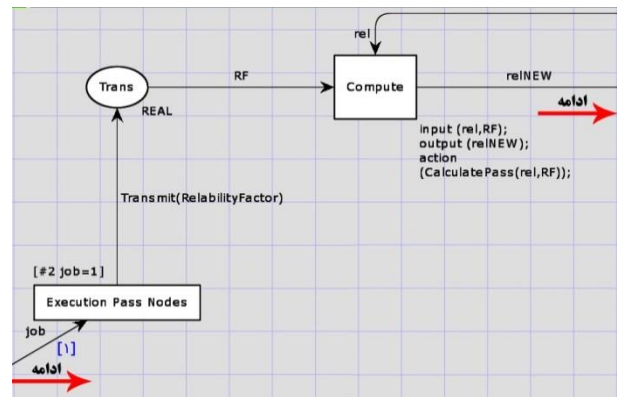
در شبکه پتری رنگی، تعداد زیادی رنگ برای شبیه سازی مدل تعریف می گردد. در جدول ۱، یک جدول از رنگ های استفاده شده و متغیرها در این پژوهش نشان داده شده است.

(جدول ۱): تمامی رنگ ها و متغیرهای استفاده شده در پیاده سازی روش پیشنهادی

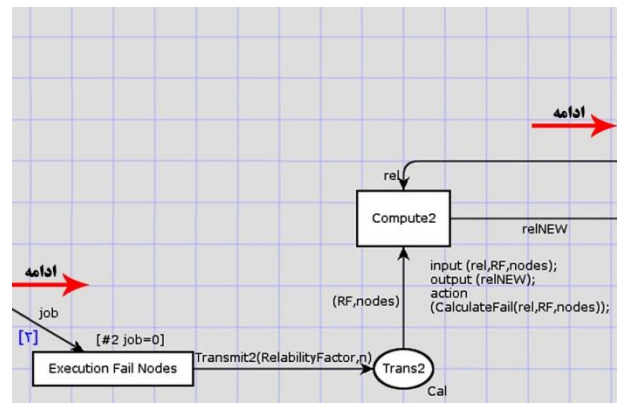
انواع closet، متغیرها، مقادیر، فرمول ها
▼ Standard declarations
▶ colset UNIT
▶ colset BOOL
▶ colset INT
▶ colset INTINF
▶ colset TIME
▶ colset REAL
▶ colset STRING
▼ colset Cloud = product INT*INT;
▼ var job:Cloud;
▼ var rel:REAL;
▼ var reINew:REAL;
▼ var RF:REAL;
▼ var nodes:REAL;
▼ var Out1:REAL;
▼ var Out2:REAL;
▼ var OutPut:REAL;
▼ colset Cal=product REAL*REAL;
▼ val reliability=1.0;
▼ val MaxReliability=1.0;
▼ val MinReliability=0.0;
▼ val ReliabilityFactor=0.1;
▼ val n=5.0;
▼ fun CalculateFail(x:REAL,y:REAL,z:REAL)= (x-(x*y*z))
▼ fun CalculatePass(x:REAL,y:REAL)= (x+(x*y))
▼ fun Transmit(RF)= (RF)
▼ fun Transmit2(RF,nodes)= (RF,nodes)
▼ fun Final(a:REAL,b:REAL,c:REAL,d:REAL)= if a+b>c then a+b else a;



(شکل ۶): توکن ها و مراحل پیاده سازی ۱



(شکل ۷): توکن ها و مراحل پیاده سازی ۲



(شکل ۸): توکن ها و مراحل پیاده سازی ۳

۰/۲	۱۹	۵	۶
۰/۰۹	۲۰	۲	۷
۰/۰۶	۱۵	۱	۸
۰/۱۷	۱۹	۴	۹
۰/۱۵	۱۶	۳	۱۰

نتایج حاصل از پیاده سازی بر روی ابرهای سری اول (تعداد کل درخواست های موجود در شبکه ۳۰ است) در جدول ۴ نشان داده شده است:

(جدول ۴-): نتایج حاصل از پیاده سازی برای تعداد ۵ ابر

شماره کار	نرخ قابلیت اطمینان	بیشترین قابلیت اطمینان	کمترین قابلیت اطمینان	پذیرش	رد	قابلیت اطمینان
۱	۰/۱	۱	۰/۵	۰	۱	۰/۹۳
۲	۰/۱	۱	۰/۵	۰	۱	۰/۸۹
۳	۰/۱	۱	۰/۵	۰	۱	۰/۸۱
۴	۰/۱	۱	۰/۵	۱	۰	۰/۸۳
۵	۰/۱	۱	۰/۵	۱	۰	۰/۹۳
۶	۰/۱	۱	۰/۵	۱	۰	۰/۹۷
۷	۰/۱	۱	۰/۵	۰	۱	۰/۹۲
۸	۰/۱	۱	۰/۵	۰	۱	۰/۸۴
۹	۰/۱	۱	۰/۵	۱	۰	۰/۸۸
۱۰	۰/۱	۱	۰/۵	۱	۰	۰/۹۴
۱۱	۰/۲	۱	۰/۵	۱	۰	۱/۰۱
۱۲	۰/۲	۱	۰/۵	۱	۰	۱/۰۷
۱۳	۰/۲	۱	۰/۵	۱	۰	۱/۱۱
۱۴	۰/۲	۱	۰/۵	۱	۰	۱/۱۸
۱۵	۰/۲	۱	۰/۵	۰	۱	۱/۱۲
۱۶	۰/۲	۱/۱	۰/۶	۱	۰	۱/۱۵
۱۷	۰/۲	۱/۱	۰/۶	۱	۰	۱/۲
۱۸	۰/۲	۱/۱	۰/۶	۱	۰	۱/۲۶
۱۹	۰/۲	۱/۱	۰/۶	۰	۱	۱/۱۹
۲۰	۰/۲	۱/۱	۰/۶	۰	۱	۱/۱۵
۲۱	۰/۳	۱/۱	۰/۶	۰	۱	۱/۱۲

۴- نتایج شبیه سازی

هر ابر حاوی سه مولفه بوده (خرابی ابر، خرابی وظیفه، خرابی ماشین مجازی) که با مقایسه آن ها می توانیم قابلیت اطمینان را ارزیابی کنیم. مقادیر وارد شده به صورت تصادفی بوده و ابرهای موجود در یک محیط محاسبات ابری را شبیه سازی می کنند. در جدول ۲، اطلاعات مربوط به ۵ ابر نشان داده شده است.

(جدول ۲-): مشخصات ابر در سری اول

شماره ابر	تعداد درخواست های رد ^{۱۲} شده	تعداد درخواست های پذیرش ^{۱۳} شده	احتمال خرابی
۱	۵	۲۰	۰/۲
۲	۴	۱۷	۰/۱۹
۳	۵	۱۹	۰/۲
۴	۳	۱۶	۰/۱۵
۵	۲	۱۸	۰/۱

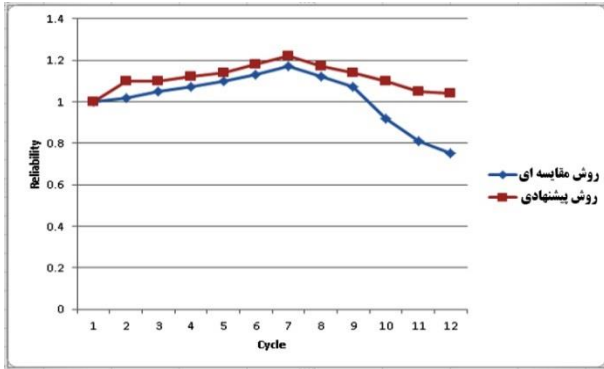
در جدول ۳، اطلاعات مربوط به ده ابر نشان داده شده است. می توان در آزمایش های بعدی از تعداد ابرهای بیشتر برای رسیدن به نتایجی پایدارتر استفاده کرد. این مقادیر نیز به صورت تصادفی ایجاد شده و محیط ابری را شبیه سازی می کند. می توان یک حد آستانه برای مقادیر ورودی هر ابر قرار داد. برای مثال برای مولفه تعداد درخواست رد شده، ماکزیمم درخواست ۵ می باشد. در جدول ۳، مقادیر مربوط به این ۱۰ ابر مورد استفاده در آزمایش نشان داده شده است.

(جدول ۳-): مشخصات ابر در سری دوم

شماره ابر	تعداد درخواست های رد شده	تعداد درخواست های پذیرش شده	احتمال خرابی
۱	۵	۲۰	۰/۲
۲	۳	۱۵	۰/۱۶
۳	۲	۱۷	۰/۱
۴	۱	۱۸	۰/۰۵
۵	۴	۱۸	۰/۱۸

¹² Pass

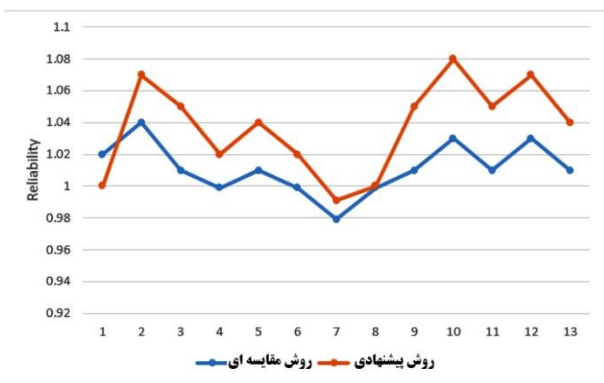
¹³ Fail



(شکل ۱۱): مقایسه قابلیت اطمینان در روش پیشنهادی و روش مقایسه ای اول

همانطور که در شکل ۱۱ مشاهده می شود، قابلیت اطمینان برای روش پیشنهادی در تکرارهای متوالی بهتر است. منظور از قابلیت اطمینان در نمودار بالا، انتخاب درست برای اجرا می باشد. در نتیجه روش پیشنهادی با در نظر گرفتن مکانیزم خطایی مذکور، می تواند درخواست ها با خطای کمتر و قابلیت بالا را انتخاب کند.

الگوریتم پیشنهادی با یک روش محاسبه میزان قابلیت اطمینان در محاسبات ابری بلادرنگ مورد مقایسه قرار گرفت [۱۸]. در شکل ۱۲، الگوریتم پیشنهادی و روش مقایسه ای از نظر میزان قابلیت اطمینان با ۱۳ کار در ۱۰ بار اجرا مورد ارزیابی و بررسی قرار گرفتند.



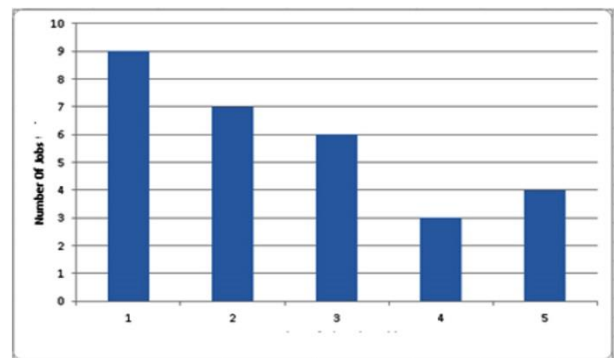
(شکل ۱۲): مقایسه قابلیت اطمینان در روش پیشنهادی و روش مقایسه ای دوم

همانطور که در شکل ۱۲ مشاهده می شود، قابلیت اطمینان برای الگوریتم پیشنهادی در تکرارهای متوالی بهتر است. منظور از قابلیت اطمینان در شکل ۱۲، انتخاب صحیح برای اجرا می باشد. در نتیجه الگوریتم پیشنهادی با در نظر گرفتن مکانیزم خطایی پیشنهادی، می تواند درخواست ها با خطای کمتر و قابلیت بالا را انتخاب کند. با توجه

۲۲	۰/۳	۱/۱	۰/۶	۱	۰	۱/۱۸
۲۳	۰/۳	۱/۱	۰/۶	۱	۰	۱/۲۲
۲۴	۰/۳	۱/۱	۰/۶	۱	۰	۱/۲۷
۲۵	۰/۳	۱/۱	۰/۶	۱	۰	۱/۳۱
۲۶	۰/۳	۱/۱	۰/۶	۱	۰	۱/۳۳
۲۷	۰/۳	۱/۱	۰/۶	۱	۰	۱/۳۸
۲۸	۰/۳	۱/۱	۰/۶	۱	۰	۱/۴۲
۲۹	۰/۳	۱/۱	۰/۶	۱	۰	۱/۴۵
۳۰	۰/۳	۱/۱	۰/۶	۱	۰	۱/۵

همانطور که در جدول ۴ مشاهده می شود با افزایش ورود وظایف پذیرش شده مقدار قابلیت اطمینان افزایش و با ورود وظایف رد شده مقدار قابلیت اطمینان کاهش می یابد.

در شکل ۱۰، نحوه استفاده از ابرها برای ۳۰ کار ورودی نشان داده شده است که در ابتدا ۱ کار رد شده و ۲۹ کار وارد شبکه ابر شده است.



(شکل ۱۰): نحوه اختصاص تعداد کارها به ابرهای مربوطه

همانطور که در شکل ۱۰ مشاهده می شود، تعداد کارهای مختلفی در ابرهای مختلف بر اساس میزان قابلیت اطمینان ابر مربوطه انجام می گردد. هر چه میزان قابلیت اطمینان ابری بالاتر باشد، میزان تعداد کارها در آن ابر، بیشتر خواهد بود.

روش پیشنهادی با یک روش محاسبه قابلیت اطمینان در محاسبات ابری بلادرنگ که توسط ملیک^{۱۴} ارائه شده مورد مقایسه قرار گرفت [۱۷].

در شکل ۱۱، روش پیشنهادی و روش مقایسه ای از نظر میزان قابلیت اطمینان در ۱۰ بار اجرا مورد ارزیابی و بررسی قرار گرفتند.

¹⁴ Malik

همچنین می توان پارامترهای جدید برای محاسبه قابلیت اطمینان ارائه کرد تا این مقدار با دقت بیشتری محاسبه گردد و نتایج بهبود یابد.

۶- مراجع

[1] جمالی، ش.، حورعلی، س. (۱۳۹۵)، «موازنه گر نامتمرکز بار در محیط ابر با بهره گیری از سیاست تصمیم گیری چندشاخصه»، مجله مهندسی برق دانشگاه تبریز، جلد ۴۶، شماره ۳، صفحه ۹۵-۱۰۶.

[2] Armbrust, M., Fox, A., Griffith, R. (2010), A view of cloud computing, *Communications of the ACM*, Vol. 53, No. 4. pp. 50-58.

[3] Barzegar, B., Motameni, H., Movaghar, A. (2019), EATSDCD: A green energy-aware scheduling algorithm for parallel task-based application using clustering, duplication and DVFS technique in cloud datacenters, *Journal of Intelligent & Fuzzy Systems* 36 (6), pp. 1-18.

[4] Jensen, K. et al. (2007), Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems, *International Journal on Software Tools for Technology Transfer* 9, pp. 213-254.

[5] Barzegar, B., Motameni, H., Bozorgi, H. (2012), Solving flexible job-shop scheduling problem using gravitational search algorithm and colored Petri net, *Journal of applied mathematics*, Volume 2012, Article ID 651310, pp. 1-20.

[6] Shahidinejad, A., Ghobaei-Arani, M., Esmaeili, L. (2020), An elastic controller using Colored Petri Nets in cloud computing environment. *Cluster Computing* Vol. 23, pp. 1045-1071.

[7] Sumangali, K., Benny, N. (2017), Advanced cloud fault tolerance system, *Materials Science and Engineering (14th ICSET)*, IOP Publishing, Volume 263, Issue 4, No. 042060, pp. 1-14.

[8] Peng, Z., Barzegar, B., Yarahmadi, M., Motameni, H., Pirouzmand, P. (2020), Energy-Aware Scheduling of Workflow Using a Heuristic Method on Green Cloud, *Scientific Programming*, Volume 2020, Article ID 8898059, pp. 1-14.

[9] Kaur, H., AL Kaur, A. (2015), A survey on fault tolerance techniques in cloud computing, *International Journal of Science, Engineering and Technology*, pp. 1-10.

[10] Anju, B., Inderveer, Ch. (2012), Fault Tolerance- Challenges, Techniques and Implementation in Cloud Computing, *IJCSI*

به نمودار پس از ورود یک وظیفه رد شده میزان کاهش قابلیت اطمینان در روش پیشنهادی کمتر از روش مقایسه ای است. در مقابل نیز پس از ورود وظیفه پذیرش شده میزان افزایش قابلیت اطمینان در روش پیشنهادی بیشتر از روش مقایسه ای می باشد. پس در نتیجه با توضیحات مذکور می توان نتیجه گرفت روش پیشنهادی نسبت به روش مقایسه ای عملکرد بهتری از خود ارائه کرده است.

۵- نتیجه گیری و پژوهش های آتی

به منظور بهبود قابلیت اطمینان در پردازش ابری تحمل پذیری خطا مورد استفاده قرار می گیرد. تحمل پذیری خطا یک خصوصیت سیستمی است که اجازه می دهد به فراهم کردن سرویس درخواست شده که احتمالاً در معرض یک خطر وجود دارد در بالاترین سطح اطمینان اجرا شود. انواع مختلف خطاها ممکن است در ساختار ابر بروز پیدا کنند و همچنین انواع مختلف تکنیک های تحمل پذیری خطا در ابر استفاده می شود بر اساس کاربرد هایی که هر کدام از این تکنیک ها می توانند داشته باشند. زمانی که یک خطا رخ می دهد این روش ها مکانیزم هایی را فراهم می کنند که سیستم نرم افزاری از وقوع خرابی اجتناب کند.

یکی از اهداف اجرای شبیه سازی تحمل پذیری خطا روی زیرساخت ابری، افزایش قابلیت اطمینان در شبکه ابری می باشد. الگوریتم پیشنهادی جهت بهبود تحمل پذیری خطا در محاسبات ابری ارائه شده است. مزایای روش پیشنهادی شامل سرعت بالا در تشخیص خطا و شناسایی آن، بهره گیری از خرابی های مهم در محاسبات ابری برای شناسایی خطا و همچنین انتخاب متوازن ماشین های مجازی برای اجرای کارها می باشد و معایب روش پیشنهادی شامل داشتن اطلاعات جامع و کافی از محیط ابر، ماشین های مجازی و کارهای ورودی، هزینه بر بودن روش پیشنهادی در بلند مدت و همچنین پیچیدگی پیاده سازی سیستم پیشنهادی می باشد.

برای کارهای آینده پیشنهاد می شود تا از الگوریتم های تکاملی همچون الگوریتم ژنتیک برای بهبود و بهینه سازی انتخاب ماشین های مجازی استفاده کرد. الگوریتم های تکاملی، جز بهترین الگوریتم های بهینه سازی محسوب می شوند و الگوریتم ژنتیک نیز پیچیدگی کمتری نسبت به بعضی از الگوریتم های تکاملی دیگر دارد و بسیار ساده و قابل فهم است. پیاده سازی آن ساده بوده و نیازی به روال های پیچیده حل مسئله ندارد.

موسسه آموزش عالی روزبهان ساری اخذ کرده است. ایشان در حال حاضر مدیرعامل شرکت دانش بنیان برنا توسعه یافته در مرکز رشد دانشگاه مازندران، کارشناس فناوری اطلاعات دفتر ویژه قوه قضاییه و همچنین مدرس دانشگاه می باشند. زمینه های پژوهشی مورد علاقه ایشان محاسبات ابری، محاسبات توزیع شده، پردازش تصویر و همچنین شبکه می باشد.

نشانه رایانامه ایشان عبارتند از:

Saed.Bahmani@gmail.com



بهنام برزگر مدرک کارشناسی خود را در رشته مهندسی کامپیوتر گرایش نرم افزار در سال ۱۳۸۵ از دانشگاه آزاد اسلامی واحد ساری، مدرک کارشناسی ارشد خود را در سال ۱۳۸۸ از دانشگاه آزاد واحد نجف آباد و مدرک دکترا را در سال ۱۳۹۷ دانشگاه آزاد واحد ساری

اخذ کرده اند. ایشان در حال حاضر به عنوان عضو هیئت علمی دانشگاه آزاد واحد بابل مشغول به کار است. زمینه پژوهشی مورد علاقه ایشان عبارتند از: محاسبات توزیع شده، محاسبات ابری، محاسبات سبز و همچنین الگوریتم های زمان بندی می باشد.

نشانه رایانامه ایشان عبارتند از:

Barzegar.Behnam@yahoo.com

روش ارجاع به مقاله: س. بهمنی، ب. برزگر. مدلسازی و ارزیابی تحمل پذیری خطا و قابلیت اطمینان در محیط ابری با استفاده از شبکه های پتری رنگی، دوفصلنامه محاسبات و سامانه های توزیع شده، سال سوم، شماره دوم، شماره پیاپی ۶، صفحه ۱ تا ۱۱، سال ۱۳۹۹.

How to cite: Saed Bahmani, Behnam Barzegar. Modeling and Evaluating Fault Tolerance and Reliability in Cloud Environment using Colored Petri Nets, Journal of Distributed Computing and Systems(JDCS), Vol 3, Issue 2, Page 1-11, 2021.

International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, pp. 288–293.

[11] متاجی خیری، خ.، مومنی، ه. ۱۳۹۴، ارائه روشی جهت افزایش تحمل خطای سیستم تولید انعطاف پذیر (FMS) با استفاده از شبکه های پتری رنگی، اولین کنگره ملی راهکارهای دستیابی به توسعه پایدار در بخشهای توسعه علم و فناوری، صفحه ۱-۱۰.

[12] Kumari, P., Kaur, P. (2018), A survey of fault tolerance in cloud computing, Journal of King Saud University – Computer and Information Sciences, pp. 1–18.

[13] Jing, Li., Yidong, Cui., Yan, Ma. (2015), Modeling Message Queueing Services with Reliability Guarantee in Cloud Computing Environment Using Colored Petri Nets, Hindawi Publishing Corporation Mathematical Problems in Engineering, Article ID 383846, pp. 1–20.

[14] شهیدی نژاد، ع. (۱۳۹۹)، مدیریت کشسانی در محاسبات ابری با استفاده از شبکه پتری رنگی، مجله مهندسی برق دانشگاه تبریز، جلد ۵۰، شماره ۳، صفحه ۱۲۶۱-۱۲۷۲.

[15] Stavros, S., Stefanos, K., Sofia, A. (2020), Modeling and Simulation of Resource Allocation Policies in Cloud Computing Using Colored Petri Nets, MDPI, Applied Sciences, Vol. 10, Issue 16, No 5644, pp. 1–23.

[16] Junna, Zh., Ao, Zh., Qibo, S., Shanguang W., Fangchun, Y. (2018), Overview on Fault Tolerance Strategies of Composite Service in Service Computing, Hindawi, Wireless Communications and Mobile Computing, Article ID 9787503, pp. 1–8.

[17] S. Malik, F. Huet. (2011), Adaptive Fault Tolerance in Real Time Cloud Computing, IEEE, World Congress on Services, Washington DC, United States, 978-0-7695-4461-8/11, pp. 280–287.

[18] AbdElfattah, E., Elkawkagy, M., El-Sisi, A. (2017), A reactive fault tolerance approach for cloud computing, 2017 13th International Computer Engineering Conference (ICENCO), 27-28 Dec., Cairo, Egypt.

سعد بهمنی مدرک کارشناسی و کارشناسی ارشد خود را در رشته مهندسی منابع طبیعی در سال ۱۳۸۹ و ۱۳۹۱ از دانشگاه آزاد اسلامی واحد قائم شهر و همچنین دومین مدرک کارشناسی ارشد خود را در رشته مهندسی فناوری اطلاعات گرایش فناوری اطلاعات در سال ۱۳۹۹ از

