



بهبود تحمل پذیری و تشخیص خطا در محاسبات ابری بر اساس دو سیستم فازی

زهرا عبادی

گروه کامپیوتر - دانشگاه آزاد اسلامی واحد بوشهر
Zahra.ebadi1398@yahoo.com

خلاصه: در سال‌های اخیر محاسبات ابری در حال تبدیل شدن به یک تکنولوژی مهم در حوزه فناوری اطلاعات است. در این تکنولوژی منابع قابل گسترش و اغلب مجازی شده، به صورت یک سرویس پردازشی و از طریق شبکه‌های ارتباطی مانند شبکه‌های محلی و اینترنت عرضه می‌شود. گاهی اوقات ممکن است، حجم و نوسانات زیاد ابر وضعیت اجرای برنامه‌ها را به یک حالت غیر منتظره هدایت کند. بنابراین برای دستیابی به قابلیت اطمینان، در دسترس بودن و بهره‌وری در محاسبات ابری، خطا و شکست باید بطور موثری پیشبینی و تشخیص داده شود. بمنظور حداقل-سازی تاثیر خرابی سیستم‌ها و اجرای صحیح برنامه‌های کاربردی از روش‌های تحمل‌پذیری خطا استفاده می‌شود. بسیاری از مطالعات تنها بر روی تشخیص خطا تمرکز دارند و بهبود تحمل‌پذیری خطا را در نظر نمی‌گیرند. به همین دلیل، در این تحقیق علاوه بر تجزیه و تحلیل دقیق ماهیت خطا و تشخیص آن، یک روش مبتنی بر فازی برای ارائه پاسخ مناسب جهت تحمل‌پذیری خطا نیز مطرح شده است. در این روش برای افزایش تحمل‌پذیری خطا و توازن بار در هنگام بروز خطا از تکنیک‌های تقاضا درخواست مجدد وظیفه و مهاجرت از طریق نقطه واریسی استفاده می‌شود. تکنیک مهاجرت با زمان، همپوشانی دارد و به همین دلیل استفاده نقطه واریسی می‌تواند تا حد ممکن از اجرای مجدد و از ابتدای وظایف پرهیز کند. نتایج روش پیشنهادی در یک دوره ۶۰ دقیقه‌ای دقت تشخیص خطا را در حدود ۹۸٪ نشان می‌دهد که به نسبت الگوریتم FLPT بطور متوسط ۴٪ برتری دارد.

کلمات کلیدی: محاسبات ابری، تشخیص خطا، تکنیک مهاجرت، نقطه واریسی، تحمل‌پذیری خطا، سیستم فازی.

۱ - مقدمه

می‌گردد [۲]. زمانی که جان مک کارتی اظهار داشت که «محاسبات ممکن است روزی به عنوان یکی از صنایع همگانی سازماندهی شود». محاسبات ابری به مفهوم فراهم کردن فناوری اطلاعات به عنوان یک صنعت همگانی شناخته می‌شود. فراهم‌کنندگان سرویس ابر مجموعه-ای از چندین مرکز داده در سراسر اینترنت دارند تا درخواست‌هایی که از سمت کاربران (از سرتاسر جهان) دریافت می‌کنند را به روش‌های بهینه سرویس‌دهی کنند. محاسبات ابری فناوری جدید برای سازمان-هایی می‌باشد که می‌توانند وظایف خود را به طور کارآمد اجرا کنند [۳]. دسته‌بندی‌های اصلی در محاسبات ابری عبارتند از: PaaS, IaaS و SaaS. IaaS شامل عرضه‌ی بسته‌ی کامل زیرساخت‌های IT مانند شبکه، سرور و فضای ذخیره‌سازی از طریق اینترنت و به شکل آنلاین

اصطلاح محاسبات ابری، بر اساس تعاریف موسسه ملی فناوری و استانداردها، اینگونه تعریف شده است: «محاسبات ابری مدلی است که برحسب تقاضای شبکه، دسترسی آسان و فراگیر به مجموعه عظیمی از منابع محاسباتی قابل تنظیم (همانند شبکه‌ها، سرورها، فضای ذخیره-سازی، برنامه‌های کاربردی و سرویس‌ها) را به سرعت و بدون دخالت سرویس دهنده به راحتی ممکن می‌سازد» [۱]. محاسبات ابری از ترکیب دو کلمه «محاسبات» و «ابر» ایجاد شده است. ابر در اینجا استعاره از شبکه‌های وسیع مانند اینترنت است که کاربر معمولی از پشت صحنه و آنچه در پی آن اتفاق می‌افتد اطلاع دقیقی ندارد (مانند داخل ابر). پیدایش مفاهیم اساسی محاسبات ابری به دهه ۱۹۶۰ باز

است. سرویس PaaS به کاربران امکان می‌دهد دغدغه‌ی نگهداری سرویس‌های زیرساختی مانند سیستم‌عامل را نداشته باشند. در سرویس SaaS، کاربر حتی با نرم‌افزار و شیوه‌ی کانفیگ آن نیز کاری نخواهد داشت. نمونه‌ی بارز SaaS استفاده از سرویس ایمیل و پست الکترونیکی است [۴].

تحمل خطای یکی از نگرانی‌های عمده برای تضمین در دسترس بودن و قابلیت اطمینان خدماتی حیاتی و همچنین اجرای برنامه‌های کاربردی است [۵]. به منظور به حداقل رساندن تاثیر خرابی بر روی سیستم و اجرای صحیح و موفق برنامه‌های کاربردی، خرابی باید پیش‌بینی شده و فعالانه مدیریت و کنترل گردد. روش‌های تحمل خطا در واقع برای پیش‌بینی این خرابی‌ها و انجام یک اقدام مناسب قبل از خرابی می‌باشد. نوع‌های مختلفی از تحمل پذیری خطا وجود دارند که یکی از آنها تحمل پذیری خطا در محاسبات ابری بلادرنگ است. سیستم‌های بلادرنگ با داشتن امتیازاتی از قبیل ابزارهای محاسبات قوی از اهمیت بسزایی برخوردارند. در محاسبات ابری بلادرنگ استفاده از نودهای پردازشگر احتمال خطا را بالا می‌برد و از طرفی چون سیستم‌های بلادرنگ از لحاظ ایمنی بحرانی هستند باید قابلیت اطمینان آنها افزایش یابد از این رو تقاضا برای دستیابی به سیستم‌هایی با قابلیت تحمل اشکال در سیستم‌های بلادرنگ رو به افزایش است [۶].

با توجه به وجود خرابی و خطاهای متنوع در محاسبات ابری، در این تحقیق فقط خطاهای مربوط به خرابی ماشین‌های مجازی و زمان اجرا بررسی می‌شوند. در خطای ماشین‌های مجازی، ممکن است یک ماشین مجازی به دلایل مختلفی از کار بیافتد. در خطاهای زمان اجرا ممکن است یک ماشین مجازی توان پاسخگویی برای اجرای دستورات یک وظیفه را نداشته باشد و یا خطای اجرای یک وظیفه به دلیل خرابی تقاضای ارسال شده از طرف کاربر باشد. در اینجا هر دو حالت خطا زمان اجرا مورد بررسی قرار خواهد گرفت. برای افزایش تحمل-پذیری خطا از تکنیک «تقاضا درخواست مجدد وظیفه»، «زمانبند واریسی مبتنی بر مهاجرت» استفاده شده است. برای این منظور موتور استنتاج فازی جهت توازن بار در هنگام بروز خطا در سیستم محاسبات ابر طراحی شده است.

۲ - پیشینه تحقیق

تحقیقات زیادی پیرامون مسئله تشخیص و تحمل‌پذیری خطا در محاسبات ابری انجام شده است که اغلب مبتنی بر سیستم‌های فازی هستند. در این بخش تعدادی از این روش‌ها بررسی می‌شوند.

در [۷] خرابی‌ها و روش‌های تحمل‌پذیری خطا با انرژی موثر را برای سیستم‌های محاسبات ابر بررسی شد. در اینجا سه روش برای این مسئله ارائه داده شده است؛ در روش اول از یک الگوریتم کارا برای تحمل‌پذیری خطا در سیستم‌های HPC استفاده شده و در روش دوم بکارگیری تکنیک مهاجرت سطح فرایند پیشگیرانه روی الگوریتم ارائه شده بررسی شده است. روش سوم مربوط به بررسی افزونگی محاسبات برای کاهش وقفه‌ها در این سیستم است. روشی برای تخصیص قابل

اطمینان منابع در رایانش ابری سیار در [۸] معرفی شد. در اینجا از یک روش نظارتی برای پیش‌بینی حالات جاری منابع براساس سابقه و ویژگی‌های دستگاه‌ها استفاده شده و از روش تاپسیس برای تخصیص رتبه اطمینان به هر دستگاه استفاده می‌شود. سپس دستگاه‌ها براساس رتبه گروه‌بندی شده و وظایف درخواستی کاربران به طور همزمان در گروه‌هایی با بالاترین رتبه اطمینان تکرار می‌شود.

تحمل پذیری خطا براساس روش VFT در رایانش ابری بررسی شد [۹]. در VFT از یک مدیر ابر و یک تصمیم‌گیرنده برای مدیریت مجازی‌سازی، توازن بار و کنترل خطاها استفاده می‌شود. در این روش ابتدا توازن بار و مجازی‌سازی انجام می‌شود و سپس تحمل‌پذیری خطا با استفاده از افزونگی، نقاط ایست‌بازرسی و کنترل‌کننده محاسبه می‌شود. احراری و همکاران، از یک رویکرد فازی برای بهبود تحمل‌پذیری خطا در پردازش ابری استفاده کردند [۱۰]. این روش بر مبنای پارامترهای منابع قوانین فازی، میزان اولویت منابع را ایجاد می‌کند که برای اینکار از ایده تعیین وزن برای دستگاه‌ها با کنترل‌کننده فازی استفاده شده است. مطهری و همکاران، روشی برای افزایش تحمل پذیری خطا در رایانش ابری ارائه دادند [۱۱]. در این روش تحمل پذیری خطا بر مبنای محاسبه نرخ موفقیت هر ماشین مجازی تضمین شده که این امر باعث کاهش قابل توجه هزینه پهنای باند مصرفی توسط کاربران می‌شود. در این روش هزینه زمانی و پردازشی برای مهاجرت از طریق نقطه واریسی کاهش می‌یابد. بدلیل همپوشانی تکنیک مهاجرت با زمان، استفاده از نقطه واریسی تا حد ممکن از اجرای مجدد و از ابتدا وظایف پرهیز می‌کند.

تشخیص خطا فازی در رایانش ابر IaaS در [۱۲] پیشنهاد شد. در این سیستم دو نوع گره محاسباتی سرورهای فیزیکی و سرورهای محاسباتی فرض شده و رسیدگی به این گره‌ها با دو مولفه «تشخیص خطا» و «مانیتورینگ» انجام می‌شود. مانیتورینگ پارامترهای شبکه نظیر زمان پاسخ، پهنای باند، توان عملیاتی، نرخ درخواست/ پاسخ و غیره را جمع‌آوری می‌کند. مولفه تشخیص خطا با توجه به روش ارائه شده به صورت منطق فازی اجرا می‌شود. علاوه بر این، به منظور اجتناب از تاثیر چند رویداد مختلف روی نتیجه استدلال، اولویت معیارها برای سرورهای فیزیکی : ۱- استفاده از منابع، ۲- توان عملیاتی، ۳- زمان پاسخ، ۴- پهنای باند. و برای سرورهای مجازی : ۱- زمان پاسخ، ۲- توان عملیاتی، ۳- پهنای باند، ۴- استفاده از منابع، فرض شده است. مالی و هوست، به تحمل خطا در رانش ابری بلادرنگ پرداختند [۱۳]. در اینجا مدل تحمل خطا برای رایانش ابری فرض شده که در آن از یک ماشین مجازی جهت جبران خطا استفاده شده است. نتایج این تحقیق نشان می‌دهد که این روش از لحاظ دقت نسبت به دو روش مورد مقایسه حدود ۱۵٪ و ۲٪ عملکرد بهتری دارد.

یک روش کارآمد برای تحمل‌پذیری بهتر خطا داده در محیط ابر در [۱۴]، پیشنهاد شد که چارچوب حمایت IDF نامیده می‌شود. در اینجا دو روش مبتنی بر الگوریتم‌های تحمل‌پذیری خطا معرفی شده که در

الگوریتم اول تحمل پذیری خطای متوسط داده درونی و در الگوریتم دوم تحمل پذیری خطای متوسط داده بیرونی ارائه شده است. روش اول تحمل خطا را درون یک وظیفه حل می کند و روش دوم تحمل خطا را در میان وظایف حل می کند. روش GACO از الگوریتم ژنتیک به منظور تحمل پذیری خطا در رایانش ابری و تصمیم گیری تخلیه بار جریان کاری استفاده می کند [۱۵]. در این روش، سرویس های تشکیل دهنده یک جریان کاری به ژن نگاشت شده و براساس اجرا روی ابر به آنها برچسب تخصیص داده می شود. مجموعه این سرویس ها یک کروموزوم را تشکیل داده که راه حل نهایی کروموزومی است که میزان برازندگی آن بر اساس زمان اجرا و انرژی مصرفی مناسب تر باشد. این تحقیق به کمک مدل تحرک RWP به پیش بینی مدت زمان نگهداری اتصال می پردازد و با لحاظ آن در محاسبه زمان و انرژی مصرفی زیان های ناشی از خطا را کاهش می دهد.

شارما و چاولا، یک روش بهینه سازی سه مرحله ای برای مهاجرت زنده ماشین مجازی مبتنی بر پیش کپی معرفی کردند [۱۶]. مهاجرت زنده ماشین مجازی یک روش حرکت ماشین مجازی در میان میزبان ها در یک مرکز داده مجازی است. مهاجرت زنده، با انتقال ماشین مجازی بدون توقف در منبع، وقفه ی سرویس را کاهش می دهد. مراحل این روش عبارتند از: ۱- کاهش انتقال صفحات حافظه در مرحله اول؛ ۲- کاهش انتقال صفحات تکراری با طبقه بندی صفحات غالباً و به ندرت به روزرسانی شده و ۳- کاهش داده های ارسال شده در آخرین تکرار مهاجرت با استفاده از تکنیک فشرده سازی RLE ساده. یک روش بهبود موازنه بار جدید در [۱۷] برای افزایش تحمل خطا و کاهش مصرف انرژی در رایانش ابری ارائه شد. این روش تکنیک تکرار وظایف برای تحمل خطا بهره می گیرد. برای اینکار سه موتور استنتاج فازی با هدف تکرار وظایف به ترتیب برای اولویت بندی به کارها، اولویت بندی مجازی و ماشین های مجازی طراحی شده است. علاوه بر این، ورودی های سیستم فازی میزان مصرف، قابلیت اطمینان و حجم کاری می باشند.

۳- روش تحقیق

کاربران از طریق رایانش ابری میتوانند تقاضاهای محاسباتی خود را از روی ماشین های مجازی آنلاین اجرا کنند. مسیرهای تخصیص منابع مورد نیاز کاربران براساس سیستم های مدیریتی و زمان بندی ابر انجام می شود. بطور کلی در یک محیط ابر N گره (سرور) فیزیکی به صورت $PS = \{PS_1, PS_2, \dots, PS_N\}$ وجود دارد که هر گره فیزیکی متشکل از یک یا چند ماشین مجازی می باشد. واحد پردازش در محاسبات ابری ماشین های مجازی (VM) می باشند. به طور کلی تعداد M ماشین مجازی به صورت $VM = \{VM_1, VM_2, \dots, VM_M\}$ در همه گره های فیزیکی وجود دارد و برای پردازش درخواست های کاربران در دسترس می باشند. علاوه بر این، تعداد R درخواست (وظیفه) بصورت $Task = \{T_1, T_2, \dots, T_R\}$ از طریق اینترنت توسط کاربران به سیستم ابر مجازی ارسال می شود. هر وظیفه شامل یک یا چند بخش

(کار) است، بنابراین $T_i = \{t_1, t_2, \dots, t_r\}$ می باشد.

در مدل پرداخت به ازای مصرف، کاربران با پرداخت هزینه بیشتر، میتوانند امکان اجرای سریع تر درخواست های خود را فراهم کنند. در رایانش ابری سیستم زمان بندی درخواست های کاربران را به گره های فیزیکی تخصیص داده و هر گره فیزیکی توسط سیستم زمان بندی داخلی خود، منابع لازم را برای اجرای درخواست های کاربران توسط ماشین های مجازی فراهم می کند. سیستم زمان بندی در رایانش ابری مسئولیت توازن بار ماشین های مجازی و کاهش زمان اجرای درخواست های کاربران را بر عهده دارد.

در صورت بروز خرابی در ماشین های مجازی یا سرورهای فیزیکی، ممکن است وضعیت اجرای برنامه های درخواستی کاربران به یک حالت غیر منتظره هدایت شود. بنابراین برای افزایش قابلیت اطمینان، بهره وری و عملکرد محاسبات ابری، خرابی و شکست باید به طور موثر تشخیص داده شود. به منظور به حداقل رساندن تاثیر خرابی بر روی سیستم ها و اجرای صحیح درخواست های کاربران از روش های تحمل پذیری خطا برای پیش بینی و انجام پاسخ مناسب استفاده می شود. سیستم های تحمل پذیری خطا در دوره های زمانی مشخصی بار ماشین های مجازی را بررسی می کنند. اگر TS کل دوره زمانی بر حسب دقیقه باشد، TS را میتوان در K دوره زمانی بصورت زیر نشان داد.

$$TS = [(t_1 - t_0), (t_2 - t_1), \dots, (t_k - t_{k-1}), \dots, (t_K - t_{K-1})] \quad (1)$$

در اینجا $(t_k - t_{k-1})$ به k -مین دوره زمانی اشاره می کند. در سال های اخیر تکنیک مهاجرت در رایانش ابری به یکی از چالش های مهم در افزایش تحمل پذیری خطا تبدیل شده است. در این تکنیک بار روی ماشین های مجازی در دوره های زمانی مختلف بررسی شده و در صورت افزایش تعداد وظایف اجرا نشده در صف یک ماشین مجازی، احتمال وجود خرابی پیش بینی شده و تعدادی وظایف از آن ماشین مجازی به سایر ماشین های مجازی انتقال می یابد. با توجه به سربار محاسباتی این تکنیک، حداقل سازی نرخ مهاجرت همراه با توازن بار مورد توجه می باشد.

در این تحقیق یک سیستم مبتنی بر فازی برای تجزیه و تحلیل و تشخیص دقیق مبدا خطا و همچنین یک سیستم مبتنی بر فازی برای ارائه پاسخ مناسب به خطای تشخیصی (سیستم تحمل پذیری خطا) پیشنهاد شده است. سیستم فازی تشخیص خطا دارای پارامترهای ورودی «زمان پاسخ»، «تراکم بار» و «توان عملیاتی» و سیستم فازی تحمل پذیری خطا دارای پارامترهای ورودی «حالت گره»، «زمان انتظار کار جاری»، «تعداد تکرار ناموفق کار جاری» و «نسبت توان عملیاتی ماشین مجازی» می باشد. هر یک از این پارامترها بصورت فازی بیان شده و استنتاج نتایج از پایگاه قوانین فازی استفاده می شوند.

به طور کلی انواع مختلفی از خطاها در رایانش ابری وجود دارد که در این تحقیق تنها خطاهای ناشی از ماشین های مجازی و زمان اجرا مورد توجه می باشد. یک ماشین مجازی ممکن است دچار خرابی نرم افزاری و یا سخت افزاری شود، لذا باید از تکنیک هایی نظیر مهاجرت تاثیر این

پارامترهای مورد نیاز سیستم فازی تشخیص خطا بطور جداگانه از هر گره فیزیکی استخراج می‌شود. هرچند اینکار بار محاسباتی و تعداد پارامترهای ورودی را افزایش می‌دهد ولی این استراتژی باعث تشخیص مبدا خطا می‌باشد، چون اطلاعات هر گره فیزیکی و توازن بار ماشین-های مجازی درون آن، در هر دوره زمانی در دسترس است. پارامترهای ورودی سیستم فازی تشخیص خطا مطابق جدول (۱) می‌باشد.

ردیف	نام پارامتر	توصیف
۱	زمان پاسخ	فاصله زمانی بین ارسال یک درخواست و ایجاد اولین پاسخ
۲	توازن بار	وظایف اختصاص یافته به یک گره فیزیکی نسبت به کل وظایف
۳	توان عملیاتی	میانگین توان عملیاتی ماشین‌های مجازی در دسترس
۴	MakeSpan	بزرگترین زمان اجرا در بین همه وظایف

$$rt_i = T_i^{response} - T_i^{sent} \quad (2)$$

توان عملیاتی از دیگر پارامترهای ورودی سیستم فازی تشخیص خطا

شکل (۱) معماری الگوریتم پیشنهادی

در سیستم پیشنهادی R درخواست توسط کاربران وارد سیستم می-شود. درخواست‌ها توسط سیستم زمانبندی محاسبات ابری به گره‌های فیزیکی تخصیص داده می‌شود. N گره فیزیکی موجود مسئولیت نگاشت و اجرای درخواست‌های کاربران را به ماشین‌های مجازی برعهده دارند. در معماری پیشنهادی مولفه مانیتورینگ به منظور استخراج پارامترهای ورودی سیستم‌های فازی در هر دوره زمانی را بر عهده دارد. این مولفه ابتدا پارامترهای ورودی سیستم فازی تشخیص خطا را براساس گره‌های فیزیکی در دسترس استخراج می‌کند. سیستم فازی تشخیص خطا براساس این پارامترها، مبدا خطای احتمالی را تشخیص داده و این اطلاعات را به سیستم تحمل‌پذیری خطا ارسال می‌کند. این سیستم از طریق مکانیسم‌های تعبیه شده، سعی در توازن بار ماشین‌های مجازی همزمان با اجرای درخواست‌های کاربران را دارد. در این تحقیق برای افزایش تحمل‌پذیری خطا از مکانیسم‌های «تقاضا درخواست مجدد وظیفه» و «مهاجرت از طریق نقطه واریسی» استفاده می‌کند. تکنیک مهاجرت با زمان همپوشانی دارد، به همین دلیل استفاده نقطه واریسی تا حد ممکن از اجرای مجدد و از ابتدای وظایف پرهیز می‌کند. در این تحقیق هر وظیفه شامل یک یا چند بخش (کار) می‌باشد. در نتیجه در لحظه بروز خطا ممکن است یک یا چند کار اجرا

است و به مفهوم ظرفیت اجرایی یک ماشین مجازی بر حسب میلیون دستورالعمل در ثانیه (MIPS) است. MakeSpan نیز به بزرگترین زمان اتمام بین همه وظایف اشاره دارد. CT_{ij} زمان اتمام وظیفه T_i روی ماشین مجازی j -ام است، CT_{max} زمان اتمام آخرین وظیفه می-باشد و بصورت زیر محاسبه می‌گردد.

$$CT_{max} = \max\{CT_{ij} \mid i \in T, i = 1, 2, \dots, R \text{ and } j \in VM, j = 1, 2, \dots, M\} \quad (4)$$

خروجی سیستم فازی تشخیص خطا با توجه به در اختیار داشتن هر یک از گره‌های فیزیکی میتواند مبدا خطا را نیز تشخیص دهد. این خروجی در سه حالت فازی «None»، «Risk» و «Danger» ایجاد یا عدم ایجاد خطا را در یک گره فیزیکی پیشبینی می‌کند. در صورت بروز خطا و با توجه به مشخص بودن گره مخرب، مولفه مانیتورینگ پارامترهای مورد نیاز سیستم تحمل‌پذیری خطا را از گره مخرب استخراج می‌کند. پارامترهای ورودی سیستم فازی تحمل‌پذیری خطا مطابق جدول (۲) می‌باشد.

جدول (۲) پارامترهای ورودی سیستم فازی تشخیص خطا

ردیف	نام پارامتر	توصیف
۱	حالت گره	خروجی سیستم فازی تشخیص خطا برای یک گره فیزیکی
۲	زمان انتظار	فاصله زمانی بین زمان پاسخ و زمان جاری سیستم
۳	تعداد تکرار ناموفق	تعداد تکرار متوالی ناموفق در اجرای یک وظیفه
۴	نرخ توان عملیاتی	نسبت مجموع توان عملیاتی ماشین‌های مجازی در دسترس به حجم کلی وظایف موجود در صف انتظار

پارامتر حالت گره خروجی سیستم فازی تشخیص خطا است و در صورت تشخیص وقوع خطا، مولفه مانیتورینگ پارامترهای زمان انتظار، تعداد تکرار ناموفق و نسبت توان عملیاتی را استخراج کرده و به سیستم فازی تحمل‌پذیری خطا ارسال می‌کند. پارامتر زمان انتظار مربوط به وظیفه جاری در حال اجرا روی گره فیزیکی مخرب است. این پارامتر میتواند برای هر سه خطا ماشین مجازی، تقاضا و زمان اجرا استفاده شود. در صورتیکه مبدا خطا درخواست تقاضا مجدد وظیفه باشد، پارامتر تعداد تکرار ناموفق میتواند از تکرار اجرای متوالی وظیفه جلوگیری کند. این پارامتر نیز برای وظیفه جاری در حال اجرا روی گره فیزیکی مخرب استخراج می‌شود. در نهایت پارامتر نرخ توان عملیاتی برای خطاهای ناشی از ماشین‌های مجازی استفاده می‌شود. این پارامتر برای گره فیزیکی مخرب بدون در نظر گرفتن ماشین‌های مجازی خراب لحاظ شده و بصورت نسبت مجموع توان عملیاتی به حجم همه وظایف در صف انتظار محاسبه می‌گردد. پارامتر نرخ توان عملیاتی از اهمیت بالایی برخوردار است، چون حتی در صورت خرابی یک ماشین مجازی ممکن است گره فیزیکی توان عملیاتی لازم را داشته باشد و نیاز به سربار محاسباتی برای انجام تکنیک‌هایی نظیر مهاجرت نباشد.

ساختار سیستم فازی پیشنهادی در مدل‌های تشخیص خطا و تحمل-پذیری خطا مشابه است. در اینجا پارامترهای ورودی از طریق مولفه مانیتورینگ استخراج شده و وارد سیستم فازی می‌شود. این پارامترها توسط مولفه فازی‌سازی از مقدار واقعی به مقدار فازی تبدیل می‌شود. پس از آن در هر دوره زمانی مشخص، موتور استنتاج فازی بررسی وقوع خطا و یا تحمل‌پذیری خطا را انجام می‌دهد. اینکار براساس پایگاه قوانین فازی انجام می‌شود که بوسیله شخص خبره ایجاد شده است. در نهایت خروجی سیستم فازی برای اعمال به محیط رایانش ابری و تبدیل به مقدار واقعی دیفازی‌سازی می‌شود. در این تحقیق برای توصیف هر یک از پارامترهای ورودی (در هر دو مدل فازی) از یک مجموعه فازی دوزنقه‌ای با تعداد پنج تقسیم‌بندی high, veryhigh, middle, low و verylow استفاده می‌شود. در اینجا از فرم قوانین فازی به صورت زیر استفاده شده است.

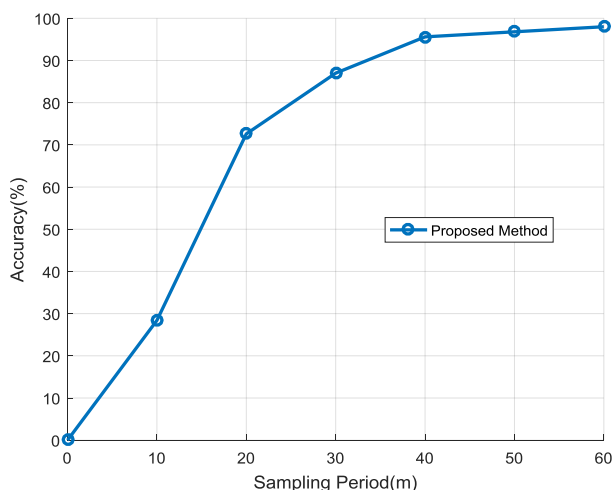
$$R_k: \text{If } (x_1 \text{ is } A_1) \text{ and } \dots \text{ and } (x_L \text{ is } A_L) \text{ Then } Y \text{ is } G \quad (5)$$

در این رابطه، R_k قانون فازی k -ام است. x_i و A_i به ترتیب به i -مین پارامتر ورودی و مجموعه فازی تخصیص داده شده به آن اشاره دارد. Y خروجی قانون و G یک ترم زبانی از مجموعه فازی متعلق به آن است. برای هر قانون یک تابع درجه عضویت مطابق پارامترهای ورودی تعریف می‌شود.

در سیستم تحمل‌پذیری خطا فازی، خروجی تقاضا درخواست مجدد وظیفه، از طریق یک پیام کنترلی به کاربر ارسال شده و درخواست جاری از سیستم زمانبندی حذف می‌گردد. در خروجی مهاجرت از طریق نقطه واریسی، اطلاعات بافر اجرای مربوط به وظیفه‌ای که دچار خطا روی ماشین مجازی شده است از طریق تکنیک مهاجرت به یک ماشین مجازی دیگری ارسال شده و ماشین مجازی جدید با توجه به اطلاعاتی که از بافر دریافت می‌کند، وظیفه را از بخش (کار) نقطه واریسی اجرا کرده و اطلاعات تکمیلی را به سیستم زمانبندی ابر ارسال می‌کند. در تکنیک مهاجرت یک یا چند وظیفه، از صف موجود در گره فیزیکی مخرب به گره فیزیکی با کمترین بار انتقال پیدا می‌کند. افزایش نرخ مهاجرت باعث افزایش زمان پاسخ برای وظایف شده و سربار محاسباتی ایجاد می‌کند، لذا باید تعداد مهاجرت‌های انجام شده حداقل شود. نرخ مهاجرت بر مبنای حجم وظایف منتقل شده نسبت به کل وظایف در یک دوره زمانی محاسبه می‌شود. برای استنتاج خروجی در هر یک از مدل‌های فازی تشخیص خطا و تحمل‌پذیری خطا، ابتدا پارامترهای سیستم محاسباتی ابری از طریق مولفه مانیتورینگ استخراج و سپس فازی‌سازی می‌شود. در مرحله بعد قانون متناسب با پارامترهای ورودی به روش ممدانی و عملگر ضرب استنتاج می‌شود.

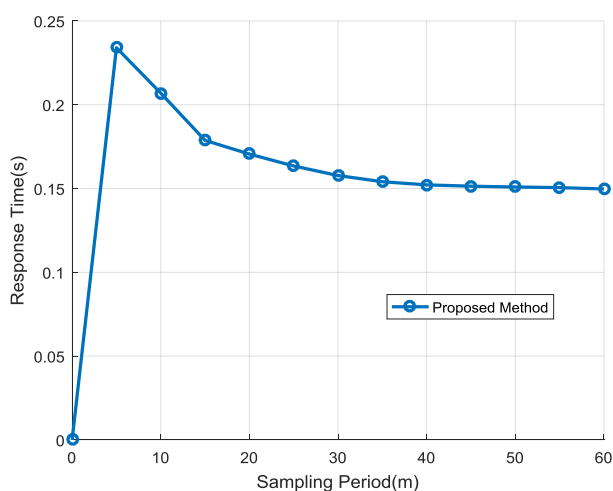
۴ - یافته ها

در این بخش آزمایش‌های گسترده‌ای را به منظور اثبات برتری روش پیشنهادی در تشخیص و تحمل‌پذیری خطا در محاسبات ابری ارائه می‌دهیم. همه آزمایش‌ها با استفاده از نرم افزار متلب ورژن ۲۰۱۷

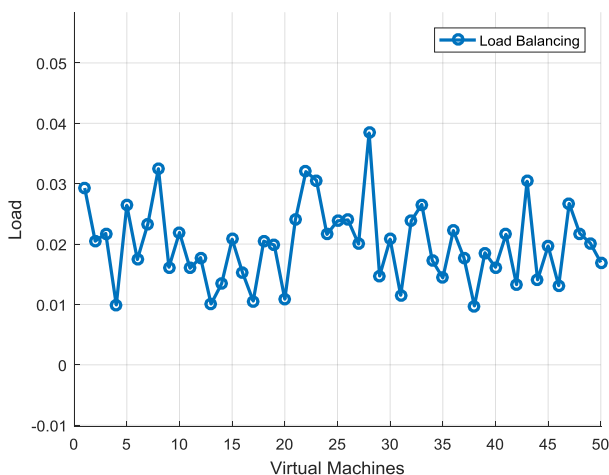


شکل (۲) نتایج بر مبنای معیار دقت تشخیص خطا

نتایج این آزمایش نشان می‌دهد که نرخ زمان پاسخ دارای نوسانات اندکی می‌باشد. نکته دیگری در مورد نتایج این آزمایش، زمان پاسخ نسبتاً بالای الگوریتم پیشنهادی در شروع شبیه‌سازی است. هر چند زمان پاسخ با افزایش دوره زمانی به کمتر از ۰/۱۴ همگرا شده است.



شکل (۳) نتایج بر مبنای زمان پاسخ



شکل (۴) توازن بار ماشین‌های مجازی در کل دوره شبیه‌سازی

انجام شده است. الگوریتم پیشنهادی با توجه به مرکز داده‌های شرکت ارتباطات ویتنام (VDC) پیاده‌سازی و آزمایش شده است [۱۲].

در شبیه‌سازی انجام شده بطور کلی تعداد ۷۵ تا ۸۵ سیستم میزبان ماشین‌های مجازی در هر زمان وجود دارد. خطاهای استفاده شده شامل سه دسته «خطا ماشین‌های مجازی»، «خطا درخواست» و «خطا زمان اجرا» با توزیع یکنواخت و با احتمال ۰/۳ به طور تصادفی در سیستم ارسال می‌شوند. ارزیابی سیستم با توجه به اعمال خطاها پس از هر دوره ۵ دقیقه محاسبه می‌شوند. نتایج شبیه‌سازی سیستم تحمل‌پذیری خطا در روش پیشنهادی و سایر روش‌های مورد مقایسه در یک دوره ۶۰ دقیقه‌ای برای ۵۰ گره فیزیکی و ۵۰ وظیفه محاسبه شده است. تعداد ماشین‌های مجازی موجود در هر گره فیزیکی به صورت تصادفی و با توزیع یکنواخت در بازه [۵ - ۱] تعیین می‌شود. همچنین توان هر ماشین مجازی برای اجرای وظایف به صورت تصادفی و با توزیع یکنواخت در بازه [۱۰۰ - ۱۰۰۰] دستورالعمل در ثانیه (MIPS) در نظر گرفته شده است. از طرفی اندازه وظایف درخواست کاربران نیز به صورت تصادفی و با توزیع یکنواخت در بازه [۲۰۰ - ۵۰۰] تعیین شده است. در نهایت هر وظیفه دارای تعداد مجزایی کار می‌باشد که در این تحقیق بازه [۱۰ - ۱] در نظر گرفته شده است.

برای ارزیابی الگوریتم‌های مختلف، سه دسته «خطا ماشین‌های مجازی»، «خطا درخواست» و «خطا زمان اجرا» بصورت تصادفی و با توجه به یکسری پارامترهای احتمالی به سیستم وارد می‌شوند. تحمل‌پذیری خطا در روش پیشنهادی با تکنیک‌های مهاجرت بر مبنای زمانبندی و ارسای و تقاضا درخواست مجدد وظیفه انجام می‌شود. نتایج شبیه‌سازی در یک دوره زمانی ۶۰ دقیقه بررسی شده و در معیار دقت تشخیص خطا و همچنین پارامترهای زمان پاسخ، تعادل بار و MakeSpan ارائه شده است. شکل (۲) نتایج را از لحاظ معیار دقت تشخیص خطا نشان می‌دهد. نتایج بر مبنای میانگین ۱۵ اجرای متمایز ارائه شده است.

نتایج این آزمایش نشان می‌دهد که روش پیشنهادی دقت تشخیص خطا مناسبی دارد. معیار دقت براساس تعداد تشخیص‌های صحیح خطا نسبت به همه خطاهای ایجاد شده محاسبه می‌شود. بهترین دقت گزارش شده توسط روش پیشنهادی در ۱۵ اجرا متوالی حدود ۹۸٪ است. در آزمایش دیگری سرعت پاسخ (زمان پاسخ) در تشخیص خطا در شکل (۳) ارائه شده است. این آزمایش نیز در یک دوره ۶۰ دقیقه‌ای شبیه‌سازی شده است.

شبیه‌سازی خطاها شامل سه دسته ماشین‌های مجازی، درخواست و زمان اجرا هستند که با احتمال $0/3$ وارد سیستم می‌شوند. ارزیابی سیستم با توجه به اعمال خطاها پس از هر دوره ۵ دقیقه محاسبه شده و نتایج در یک دوره ۶۰ دقیقه‌ای برای ۵۰ گره فیزیکی و ۵۰ وظیفه گزارش می‌شود. برای ارزیابی از دقت تشخیص خطا، زمان پاسخ، توازن بار و MakeSpan استفاده شده است. در سیستم فازی تشخیص خطا از پارامترهای زمان پاسخ، توازن بار، توان عملیاتی و MakeSpan به منظور بررسی وضعیت ماشین‌های مجازی و عملکرد آنها استفاده شد. پیشنهاد می‌شود برای کارهای آینده کارایی پارامترهای دیگری نظیر پهنای باند، منابع در دسترس، بهره وری و غیره نیز در تشخیص خطا ارزیابی شوند.

مراجع

- [1] Amin, Z., Singh, H., & Sethi, N. (2015). Review on fault tolerance techniques in cloud computing. *International Journal of Computer Applications*, 116(18).
- [2] Cheraghlo, M. N., Khadem-Zadeh, A., & Haghparast, M. (2016). A survey of fault tolerance architecture in cloud computing. *Journal of Network and Computer Applications*, 61, 81-92.
- [3] Costa, B. S. J., Angelov, P. P., & Guedes, L. A. (2015). Fully unsupervised fault detection and identification based on recursive density estimation and self-evolving cloud-based classifier. *Neurocomputing*, 150, 289-303.
- [4] Osanaiye, O., Chen, S., Yan, Z., Lu, R., Choo, K. K. R., & Dlodlo, M. (2017). From cloud to fog computing: A review and a conceptual live VM migration framework. *IEEE Access*, 5, 8284-8300.
- [5] Singh, S., & Chana, I. (2016). QoS-aware autonomic resource management in cloud computing: a systematic review. *ACM Computing Surveys (CSUR)*, 48(3), 42.
- [6] Cheraghlo, M. N., Khademzadeh, A., & Haghparast, M. (2019). New Fuzzy-Based Fault Tolerance Evaluation Framework for Cloud Computing. *Journal of Network and Systems Management*, 1-19.

[۷] بیانگر، کاظم، بارانی، هادی. (۱۳۹۷). بررسی خرابی‌ها و روش‌های تحمل‌پذیری خطا با انرژی مؤثر برای سیستم‌های محاسبات با کارایی بالا در محیط ابری. دومین همایش ملی دانش و فناوری مهندسی برق. کامپیوتر و مکانیک ایران، تهران. موسسه برگزار کننده همایش‌های توسعه محور دانش و فناوری سام ایرانیان.

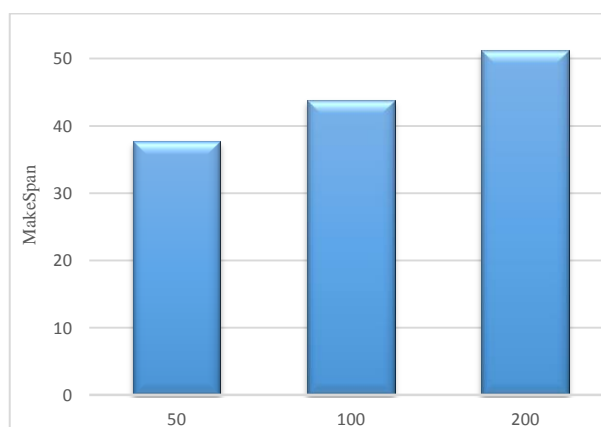
[۸] نجف‌آبادی‌سامانی، زهرا، خیام‌باشی، محمدرضا. (۱۳۹۵). ارائه‌ی روشی برای تخصیص قابل اطمینان منابع در رایانش ابری سیار. هشتمین کنفرانس بین‌المللی فناوری اطلاعات و دانش. همدان. انجمن فناوری اطلاعات و ارتباطات ایران. دانشگاه بوعلی سینا.

[۹] آرمانده، روشنگر، حنائی، علی. (۱۳۹۵). بررسی تحمل‌پذیری خطا براساس روش VFT در رایانش ابری. اولین همایش ملی نگرشی نوین در مهندسی برق و کامپیوتر. کرمانشاه، دانشگاه آزاد اسلامی واحد کرمانشاه.

[۱۰] احراری، محمدجواد، حسنی‌آهنگر، محمدرضا، غفوری، آرش. (۱۳۹۷). ارائه یک روش زمان‌بندی وظیفه تحمل‌پذیر خطا برای ماشین‌های مجازی در محیط محاسبات ابری. پنجمین کنفرانس بین‌المللی

در آزمایش بعدی وضعیت بار ماشین‌های مجازی بطور متوسط در کل دوره‌های زمانی محاسبه و در شکل (۴) نشان داده شده است. نتایج به وضوح نشان می‌دهد که بار اغلب ماشین‌های مجازی در یک سطح است و نوسانات کمی دارد. بار همه ماشین‌های مجازی تقریباً در محدوده بین $0/01$ تا $0/03$ در نوسان است. این نتایج عملکرد مناسب روش پیشنهادی را در توازن بار سیستم نشان می‌دهد.

در آزمایش دیگری روش‌های مختلف از لحاظ MakeSpan مورد ارزیابی قرار داده شده است. شکل (۵) نتایج این آزمایش را برای روش پیشنهادی نشان می‌دهد. نتایج بر مبنای یک دوره زمانی ۶۰ دقیقه‌ای با تعداد وظایف متفاوت ۵۰، ۱۰۰ و ۲۰۰ محاسبه شده است.



شکل (۵) نتایج روش پیشنهادی از لحاظ MakeSpan

۵ - نتیجه‌گیری

گاهی اوقات، حجم کار نوسانی و زیاد محاسبات ابری به نحوی است که ممکن است وضعیت اجرای برنامه‌ها به یک حالت غیر منتظره هدایت شود. برای دستیابی به در دسترس بودن، عملکرد، استحکام و قابلیت اطمینان در محاسبات ابری، خطا و شکست باید به طور مؤثری مورد دستیابی قرار گیرد و مشخص شود. انواع خطاها را میتوان با توجه به موقعیت‌هایی که در آن وجود دارد طبقه‌بندی کرد. این خطاها میتواند از زیرساخت‌های فیزیکی (خرابی گره فیزیکی)، از سیستم مجازی سازی (خرابی ماشین مجازی) یا مدیر زمانبندی (خرابی سیستم زمانبند) ناشی شود. علاوه بر این، خطا میتواند از خرابی موجود در درخواست کاربران ناشی شود. برای ایجاد یک راه حل مؤثر در مقابله با خطاها، باید انواع خطاهای احتمالی جمع‌آوری و شناسایی شوند تا بتوان مراحل پاسخ مربوطه را ارسال کرد. با این حال، بسیاری از مطالعات بر بهبود تشخیص خطا در محاسبات ابری تمرکز ندارند و تنها برای تشخیص خطا معرفی شده‌اند. به همین دلیل، هدف از این مقاله، اول تجزیه و تحلیل راه‌حل‌های فعلی برای رفع مشکلات آنها و تشخیص بهتر خطا می‌باشد. پس از آن، یک الگوریتم مبتنی بر منطق فازی و تکنیک افزایش تحمل‌پذیری خطا پیشنهاد داده شد که زودتر خطاها را تشخیص و رفع کند.

شبیه‌سازی با توجه به مرکز داده‌های VDC استفاده شده است. در

فناوری اطلاعات کامپیوتر مخابرات. تفلیس -کشور گرجستان،
دبیرخانه دائمی کنفرانس.

[۱۱] مطهری، شیما، فاضلی، مهدی، پاطوقی، احمد. (۱۳۹۴). ارائه راهکاری
برای افزایش قابلیت اطمینان در رایانش ابری. هفتمین کنفرانس بین
المللی فناوری اطلاعات و دانش. ارومیه. دانشگاه ارومیه.

- [12] Bui, D. M., & Lee, S. (2018). Early fault detection in IaaS cloud computing based on fuzzy logic and prediction technique. *The Journal of Supercomputing*, 74(11), 5730-5745.
- [13] Malik, S., & Huet, F. (2011, July). Adaptive fault tolerance in real time cloud computing. In *Services (SERVICES), 2011 IEEE World Congress on* (pp. 280-287). IEEE.
- [14] Song, B., Ren, C., Li, X., & Ding, L. (2014, September). An efficient intermediate data fault-tolerance approach in the cloud. In *2014 11th Web Information System and Application Conference* (pp. 203-206). IEEE.
- [15] Deng, S., Huang, L., Taheri, J., & Zomaya, A. Y. (2014). Computation offloading for service workflow in mobile cloud computing. *IEEE Transactions on Parallel and Distributed Systems*, 26(12), 3317-3329.
- [16] Sharma, S., & Chawla, M. (2016). A three phase optimization method for precopy based VM live migration. *SpringerPlus*, 5(1), 1022.
- [17] Moghtadaeipour, A., & Tavoli, R. (2015, November). A new approach to improve load balancing for increasing fault tolerance and decreasing energy consumption in cloud computing. In *2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEL)* (pp. 982-987). IEEE.