دانشگاه شیخ بهائی

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

پایان نامه کارشناسی ارشد رشته مهندسی کامپیوتر- نرم افزار

ارائه روشی برای جلوگیری از گسترش خطا در یک محیط توزیع‌شده

پژوهشگر

یاسر وکیلی

استاد راهنما

دکتر محمدرضا خیام‌باشی

مهر ۱۳۹۰

تقدیم

تقدیم به پدر و مادر

نهال را باران باید تا بشوید غبار نشسته بر برگ‌هایش

و سیرابش کند از آب حیات و آفتاب باید

تا بتاباند نیرو را

و محکم کند شاخه‌های تازه روییده را

به نام مادر

بوسه‌ای باید زد دست‌هایی را

می‌شویند غبار خستگی روزگار را و سیراب می‌کنند روح تشنه را

به نام پدر

بوسه‌ای باید زد دست‌هایی را

که می‌تابانند نیرو را

و محکم می‌کنند استواری پایه‌های زیستن را

سپاس‌گزاری

با درود فراوان به روح پر فتوح پدر بزرگوارم و سپاس بیکران بر همدلی و همراهی و همگامی مادر دلسوز و مهربانم که سجده‌ی ایثارش گل محبت را در وجودم پروراند و دامان گهربارش لحظه‌های مهربانی را به من آموخت.

و با تقدیر و تشکر شایسته از استاد فرهیخته و فرزانه جناب آقای دکتر محمدرضا خیام‌باشی که با نکته‌های دلاویز و گفته‌های بلند، صحیفه‌های سخن را علم‌پرور نمود و همواره راهنما و راه‌گشای نگارنده در اتمام و اکمال پایان‌نامه بوده است.

معلما مقامت ز عرش برتر باد همیشه توسن اندیشه‌ات مظفر باد

و در پایان بسی شایسته است از آقایان جناب آقای دکتر احمد براآنی دستجردی و جناب آقای دکتر بهمن زمانی که قبول زحمت فرموده و با کرامتی چون خورشید، سرزمین دل را روشنی بخشیدند و گلشن سرای علم و دانش را با راهنمایی‌های کارساز و سازنده بارور ساخته و داوری این پایان‌نامه را قبول کردند، تقدیر و تشکر می‌نمایم.

چکیده

سیستم‌های توزیع شده[1] یکی از پر اهمیت‌ترین زمینه‌های پژوهشی در دانش‌های رایانه[2] می‌باشند. در این محیط برای رسیدن به دسترس‌پذیری بالای داده و قابلیت اطمینان، از شگرد تکرار داده بهره برده می‌شود. یکی از ویژگی‌های با ارزش این شگرد، یکسان بودن تکرارها از دیدگاه نرم‌افزاری می‌باشد. از سوی دیگر کمبودهای نرم‌افزاری رخداد خطاها را در داده‌های تکرار شده به‌همراه خواهد داشت. بنابراین می‌توان گفت اگر خطایی در یکی از این تکرارها رخ بدهد آن‌گاه احتمال رخداد همان خطا تحت آن شرایط در بقیه تکرارها نیز وجود خواهد داشت. حال اگر یک خطا به صورت تکراری در داده‌های تکرار رخ دهد، گسترش خطا در سیستم پدید می‌آید. روی هم‌رفته رخداد هر خطا در سیستم هزینه تحمل‌پذیری خطا را در پی خواهد داشت. مسئله بنیادی رخداد پدیده گسترش خطا در سیستم می‌باشد. بنابراین این پدیده هزینه تحمل‌پذیری خطای کل سیستم را به‌صورت فزاینده افزایش خواهد داد. در این پایان‌نامه روشی برای جلوگیری از رخداد پدیده گسترش خطا ارائه شده است. در این روش کارشناس تشخیص خطا به عنوان یک نگرنده در نقطه‌های مستعد خطا قرار داده می‌شود. این نگرنده با وارسی درخواست‌های در حال انجام در نقطه‌های مستعد خطا، خطاهای رخداده را شناسایی کرده و در جدول ویژگی‌های خطا ذخیره می‌کند. در گام پسین اگر همان درخواست دوباره فرستاده شود آن‌گاه درخواست مورد نظر پردازش نشده، از رخداد دوباره خطا جلوگیری خواهد شد. دستاوردهای ارزیابی نشان می‌دهد که نرخ رخداد خطای تاثیر گرفته شده، هزینه تحمل‌پذیری خطا، در هنگام رخداد پدیده گسترش خطا به صورت چشمگیری بهبود یافته است.

کلیدواژه: سیستم‌های توزیع شده، تکرار داده، خطا، تحمل‌پذیری خطا، گسترش خطا

[1] Distributed systems
[2] کامپیوتر

فهرست نوشته‌ها

دیباچه ۱۱

۱-۱- پیش‌گفتار ۱۱

۱-۲- طرح مسئله ۱۲

۱-۳- هدف پژوهش ۱۲

۱-۴- محدوده پژوهش ۱۳

۱-۵- مراحل انجام پژوهش ۱۳

۱-۶- ساختار پایان‌نامه ۱۴

فصل دوم ۱۵

دانسته‌های پایه‌ای و پیشینه پژوهش ۱۵

۲-۱- پیش‌گفتار ۱۵

۲-۲- سیستم‌های توزیع شده ۱۵

۲-۳- هدف‌های موجود در سیستم‌های توزیع شده ۱۶

۲-۳-۱- دسترس‌پذیری به داده‌ها یا منبع‌ها ۱۶

۲-۳-۲- شفافیت پخش ۱۷

۲-۳-۳- باز بودن ۱۸

۲-۳-۴- توسعه‌پذیری ۱۸

۲-۴- تکرار داده ۱۸

۲-۴-۱ تکرار داده در سطح دسترسی ۱۹

۲-۴-۲ تکرار داده در سطح فرایند ۱۹

۲-۴-۳- الگوریتم سازگاری تکرار داده پویا ۲۲

۲-۵- تشخیص و شناسایی خطا ۲۲

۲-۵-۱- شناسایی خطا با بهره از دستور زبان سیستم‌های نرم‌افزاری ۲۳

۲-۵-۲- شناسایی خطا با بهره از تحلیل افزونگی فازی ۲۵

۲-۵-۳- یک معماری برای شناسایی و جداسازی خطا بر پایه متدهای فازی ۲۸

۲-۵-۴- آتاماتای فازی برای شناسایی خطا ۳۱

۲-۵-۵- پروتکل شناسایی خطای توزیع شده برای شبکه‌های بدون برنامه‌ریزی‌شده سیار و بی‌سیم ۳۵

۲-۵-۶- یک توافق برای رسیدن به شناسایی خطا بر روی یک شبکه کلی غیرقابل اعتماد ۳۶

۲-۵-۷- یک رهیافت برای شناسایی، جداسازی و راه‌اندازی خطاهای عمل‌کننده در سیستم‌های کنترل با قابلیت پیشبینی دارای مدل توزیع شده ۳۶

۲-۵-۸- یک رهیافت برای مسئله‌های شناسایی، جداسازی خطاهای حسگر و عمل‌کننده برای یک کلاس از سیستم‌های غیرخطی ۳۶

۲-۶- تحمل‌پذیری خطا ۳۶

۲-۶-۱- تحمل‌پذیری خطا با افزونگی ۳۷

۲-۶-۲- الگوریتم تحمل‌پذیری خطا برای تکرار داده به صورت پویا در سیستم‌های توزیع شده ۳۸

۲-۶-۳- طرح پرهیز از خطا در سیستم‌های تکرار آنی ۳۹

۲-۶-۴- یک معماری برای تحمل‌پذیری خطای نرم‌افزاری با بهره از انتزاع‌ها ۴۳

۲-۶-۵- یک تئوری برای تحمل‌پذیری خطای بر پایه مشاهده ۴۴

۲-۶-۶- یک متد برای مدل کردن تحمل‌پذیری خطا در سیستم‌های توزیع شده برای ترکیب کنترلگرهای مجزا ۴۴

۲-۶-۷- تحلیل تحمل‌پذیری خطا و قابلیت اطمینان در معماری‌های سیستم زمان واقعی توزیع شده ۴۴

۲-۶-۸- یک رهیافت سازگار برای به‌دست‌آوردن تحمل‌پذیری خطای نرم‌افزاری و سخت‌افزاری در محیط دارای محاسبه توزیع شده ۴۴

۲-۶-۹- یک متد گروهی حد نصاب توزیع شده کامل با تحمل‌پذیری بالا و سربار ارتباطی پایین ۴۵

۲-۶-۱۰- یک متد برای تحمل‌پذیری موثر شکست‌ها در سیستم‌های توزیع شده زمان واقعیغیرهمزمان ۴۵

۲-۶-۱۱- یک الگوریتم برای هماهنگی نقطه بررسی کنترل‌شده در برنامه کاربردی برای تحمل‌پذیری خطا در سیستم‌های توزیع شده دارای محاسبه ۴۵

۲-۶-۱۲- یک طراحی و پیاده‌سازی از ساختار تحمل‌پذیری خطای بیزانتین برای وب‌سرویس‌ها ۴۶

۲-۶-۱۳- یک رهیافت تحمل‌پذیری خطا بر پایه الگوریتم برای قابلیت اجرایی بالای محاسبه‌ها ۴۶

۲-۶-۱۴- یک رهیافت برای محاسبه تحمل‌پذیری خطای توسعه چندعاملی‌ها ۴۶

۲-۷- جمع‌بندی ۴۷

فصل سوم ۴۸

بیان روش پیشنهادی ۴۸

۳-۱- پیش‌گفتار ۴۸

۳-۲- ساختار ارتباطی سیستم ۴۹

۳-۳ ساختار سیستم در سطح سیستم عامل ۵۰

۳-۴ ساختار سیستم برای بالابردن دسترس‌پذیری بر روی داده ۵۱

۳-۵ ساختار سیستم برای تحمل‌پذیری خطا ۵۲

۳-۶- اثبات افزایش هزینه تحمل‌پذیری خطای سیستم با رخداد گسترش خطا ۵۳

۳-۷- تولید رزیدوآل ۵۴

۳-۸- کارشناس تشخیص خطا ۵۶

۳-۹- استقرار کارشناس تشخیص خطا بر روی سیستم ۵۷

۳-۱۰- مولفه‌های کارشناس تشخیص خطا ۵۸
۳-۱۰-۱- مولفه بررسی کننده خطا ۵۸
۳-۱۰-۲- مولفه شناسایی کننده ۵۹
۳-۱۰-۳- مولفه انتخاب کننده ۶۰
۳-۱۰-۴- مولفه انبار الگوها ۶۱
۳-۱۰-۵- مولفه ارسال کننده ۶۶
۳-۱۰-۶- مولفه تولید رزیدوآل ۶۶
۳-۱۰-۷- مولفه تحمل‌پذیری خطا ۶۶
۳-۱۱- جدول‌های پایگاه‌داده‌ای کارشناس تشخیص خطا ۶۷
۳-۱۱-۱- جدول ویژگی‌های خطا ۶۷
۳-۱۱-۲- جدول ویژگی‌های کار ۶۸
۳-۱۲- استقرار جدول‌های پایگاه‌داده‌ای بر روی سیستم توزیع شده ۷۰
۳-۱۲-۱ قرار دادن جدول ویژگی‌های خطا بر روی همه ماشین‌ها ۷۱
۳-۱۲-۲ قرار دادن جدول ویژگی‌های خطا بر روی یک ماشین مرکزی ۷۲
۳-۱۳- میزان حافظه مورد نیاز جدول‌های ویژگی‌های خطا و ویژگی‌های کار بر روی هر ماشین ۷۳
۳-۱۴- محاسبه هزینه بروزرسانی جدول ویژگی‌های خطا بر روی ماشین‌ها ۷۵
۳-۱۵- عملکرد کارشناس تشخیص خطا ۷۷
۳-۱۶- ویژگی‌های کلی روش ارائه شده ۷۹
۳-۱۷- جمع‌بندی ۸۰
فصل چهارم ۸۱
ارزیابی روش پیشنهادی ۸۱

۴-۱- پیش‌گفتار ۸۱
۴-۲- نرخ رخداد خطای تاثیر گرفته شده ۸۲
۴-۳- هزینه تحمل‌پذیری خطا در هنگام رخداد خطاهای تکراری ۸۵
۴-۴- ارزیابی سیستم‌ها از نقطه‌نظر هزینه تحمل‌پذیری خطا در هنگام رخداد خطاهای گوناگون ۸۶
۴-۵- مقایسه روش مورد نظر با روش‌های ارائه شده ۸۶
۴-۶- سربار زمانی ۸۸
۴-۶-۱- مطالعه موردی برای محاسبه سربار زمانی ۸۸
۴-۶-۲- محاسبه سربار زمانی کارشناس تشخیص خطا پس از فعال‌شدن ۹۰
۴-۶-۳- مقایسه سربار زمانی تحمیل‌شده پیش از فعال‌شدن کارشناس تشخیص خطا ۹۲
۴-۶-۴- مقایسه سربار زمانی تحمیل‌شده پس از فعال‌شدن کارشناس تشخیص خطا تا شناسایی رخداد نخستین خطا ۹۲
۴-۷- جمع‌بندی ۹۴
فصل پنجم ۹۶
خلاصه و نتیجه‌گیری ۹۶
۵-۱- جمع‌بندی پایانی ۹۶
۵-۳- کارهای آینده ۹۸
منبع‌ها ۹۹

فهرست شکل‌ها

شکل ۱-۱ نمایی از ساختار گام‌های انجام پایان‌نامه ۱۴

شکل ۲-۱ تکرار غیر فعال ۲۰

شکل ۲-۲ تکرار فعال ۲۱

شکل ۲-۳ تکرار نیمه غیر فعال ۲۲

شکل ۲-۴ تولید رزیدوآل در سیستم‌های با رفتار خطی ۲۷

شکل ۲-۵ نمای کلی تولید و ارزیابی رزیدوآل ۲۸

شکل ۲-۶ گام‌های شناسایی و جداسازی خطا ۲۹

شکل ۲-۷ نمایش مقدار تابع‌های عضویت به صورت ذوزنقه‌ای ۳۱

شکل ۲-۸ حالت‌ها و کمان‌های آتاماتا برای ترسیم یک سیستم رویداد گسسته ۳۲

شکل ۲-۹ آتاماتای فازی ۳۴

شکل ۲-۱۰ نمایی از افزونگی فیزیکی ۳۸

شکل ۲-۱۲ نمایی از کارایی الگوریتم FDR ۳۹

شکل ۲-۱۳ نمای بهره از نرم‌افزارها و سخت‌افزارهای گوناگون ۴۰

شکل ۲-۱۴ ناحیه‌های شکست دو ماشین به همراه ناحیه شکست مشترک ۴۱

شکل ۳-۱ نمایی از یک سیستم توزیع شده با ساختار ارتباطی درختی ۴۹

شکل ۳-۲ نمایی از سیستم عامل توزیع شده آموابا ۵۱

شکل ۳-۳ نمایی از طرح تکرار ایستا ۵۲

شکل ۳-۴ نمای کلی تولید رزیدوآل در روش مورد نظر ۵۵

شکل ۳-۵ عملکرد کارشناس تشخیص خطا به صورت نگرنده ۵۷

شکل ۳-۶ استقرار کارشناس تشخیص خطا به صورت نگرنده بر روی ماشین‌ها ۵۸
شکل ۳-۷ شبه کد مولفه بررسی کننده خطا ۵۹
شکل ۳-۸ شبه کد مولفه شناسایی کننده ۶۰
شکل ۳-۹ شبه کد مولفه انتخاب کننده ۶۱
شکل ۳-۱۰ ساختار داخلی انبار الگوها ۶۲
شکل ۳-۱۱ دسته‌بندی راه‌حل‌های پیاده‌سازی شده در مولفه‌های الگوهای ثبت شده و شبه‌الگوها ۶۴
شکل ۳-۱۲ شبه کد انبار الگوها ۶۵
شکل۳-۱۳ جدول ویژگی‌های خطا ۶۸
شکل ۳-۱۴ جدول ویژگی‌های کار ۶۹
شکل ۳-۱۵ استقرار جدول ویژگی‌های خطا بر روی همه ماشین‌ها ۷۱
شکل ۳-۱۶ استقرار جدول ویژگی‌های خطا بر روی ماشین مرکزی ۷۳
شکل ۳-۱۷ جدول ویژگی‌های مربوط به فضای مصرفی جدول‌های پایگاه‌داده‌ای ۷۴
شکل ۳-۱۸ شبه کد به‌دست آوردن فضای مورد نیاز جدول‌های بر روی هر ماشین ۷۵
شکل ۳-۱۹شبه کد درج ویژگی‌های خطا ۷۶
شکل ۳-۲۰ شبه کد عملکرد کارشناس تشخیص خطا ۷۸
شکل ۴-۱ ناحیه‌های رخداد خطاهای ماشین‌ها ۸۲
شکل ۴-۲ ارزیابی نرخ رخداد خطای تاثیر گرفته شده ۸۴
شکل ۴-۳ هزینه تحمل‌پذیری خطا در هنگام رخداد خطاهای تکراری ۸۵
شکل ۴-۴ هزینه تحمل‌پذیری خطا در هنگام رخداد خطاهای گوناگون ۸۶
شکل ۴-۵ شبه کد شبه الگوی تولید شماره حساب ۸۹
شکل ۴-۶ سربار زمانی نظارت بدون شناسایی خطای کارشناس تشخیص خطا در اجراهای پی‌درپی ۹۱

شکل ۴-۷ سربار زمانی سیستم پیش از فعال‌شدن کارشناس تشخیص خطا ۹۲

شکل ۴-۸ سربار زمانی سیستم پس از فعال‌شدن تا شناسایی رخداد خطا ۹۳

فهرست جدول‌ها

جدول ۲-۱ نرخ‌های شکست بر اساس گوناگونی ۴۳

جدول ۳-۱ کارایی ارتباطات آمواپا ۷۶

جدول ۴-۱ مقدارهای نرخ‌های رخداد خطا ۸۳

جدول ۴-۲ مقدارهای نرخ رخداد خطای تاثیر گرفته شده ۸۳

جدول ۴-۳ مقایسه رهیافت‌ها از دیدگاه شناسایی خطا ۸۷

جدول ۴-۴ سربار زمانی اجرای درخواست‌ها ۸۹

جدول ۴-۵ سربار زمانی کارشناس تشخیص خطا ۹۰

جدول ۴-۶ سربار زمانی نظارت بدون شناسایی خطای کارشناس تشخیص خطا در اجراهای پی‌درپی ۹۱

فهرست معادله‌ها

معادله (۲-۷): ورودی-خروجی سیستم ۲۶

معادله (۲-۸): ورودی-خروجی سیستم به همراه شرایط‌های رخداد خطا ۲۶

معادله (۲-۹): بردار رزیدوآل در سیستم‌های با رفتار خطی ۲۶

معادله (۲-۱۰): اثرهای ورودی‌های ناشناخته ۲۶

معادله (۲-۱۱): مدل فازی ۲۷

معادله (۲-۱۱): تابع فازی سوگنو ۲۷

معادله (۲-۱۲): بردار رزیدوآل در سیستم‌های غیر خطی ۲۸

معادله (۲-۱۳): مقدار رزیدوآل برای شناسایی خطا ۳۰

معادله (۲-۱۴): مقدار رزیدوآل برای ارزیابی خطا ۳۰

معادله (۲-۱۵): مقدار رزیدوآل دقیق‌تر برای ارزیابی خطا ۳۰

معادله (۲-۱۶): جداسازی خطا ۳۱

معادله (۲-۱۷): مجموعه خطاها ۴۱

معادله (۲-۱۸): نرخ شکست مشترک ۴۲

معادله (۳-۱): مجموعه درخواست‌های فرستاده شده کاربر بر روی ماشین‌ها ۵۰

معادله (۳-۲): طرح تکرار ایستا ۵۱

معادله (۳-۳): مجموعه خطاهای رخ داده بر روی ماشین‌ها ۵۳

معادله (۳-۴): تکرار رخداد یک خطا بر روی ماشین‌ها ۵۳

معادله (۳-۵): هزینه تحمل‌پذیری خطا ۵۳

معادله (۳-۶): هزینه پدیده گسترش خطا ۵۳

معادله (۳-۷): تولید رزیدوآل ۵۵

معادله (۳-۸): هزینه بروزرسانی ۷۷

فصل اول
دیباچه

۱-۱- پیش‌گفتار

به طور کلی اینترنت، به سرعت ما را به سمت پخش‌شدگی یا محیط اطلاعاتی به هم پیوسته برده است. در این محیط، داده‌ها در جاهای جغرافیایی گوناگون از هم قرار دارند. این ویژگی انگیزه خواندن و نوشتن داده‌ها را به صورت محلی پدید آورده است. در نتیجه دسترس‌پذیری بر روی داده بالارفته و در پایان قابلیت اجرایی سیستم به صورت جدی افزایش یافته است [۱]. برای رسیدن به این هدف شگردی به نام تکرار داده[1] به‌کار گرفته شده است.

از سوی دیگر کمبودهای طراحی و شکاف‌های امنیتی، خطاهایی را در نرم‌افزار به همراه داشته که در پایان این خطاها به شکست‌هایی در کل سیستم منجر خواهند شود. حال اگر خطایی در یکی از تکرارهای داده رخ دهد، به احتمال زیاد این خطا در تکرارهای دیگر نیز رخ خواهد داد.

در کل برای به دست‌آوردن سیستم‌هایی با قابلیت اطمینان بالا از دو روش مکمل به نام‌های پیش‌گیری از خطا و تحمل‌پذیری خطا به کار گرفته می‌شوند. روش پیش‌گیری از خطا، خطاها را در زمان طراحی و پیاده‌سازی و پیش از نمایان شدن در محیط کاری (زمان سرویس‌دهی و اجرای سیستم) شناسایی و حذف می‌کند.

به شناسایی خطاها در گام طراحی و پیاده‌سازی سیستم‌ها پرهیز از خطا گفته می‌شود. به از بین بردن خطاهای دیده شده در گام آزمایش و خطایابی از سیستم‌ها حذف خطا گفته می‌شود. پس می‌توان گفت روش پیش‌گیری از

[1] Data replication

خطا با این دو گام کاری قابلیت اطمینان سیستم را افزایش خواهد داد. ولی در کل این روش، روش کارایی برای به‌دست آوردن کارهای با قابلیت اطمینان بالا نیست. زیرا مولفه‌های نرم‌افزاری و سخت‌افزاری درون سیستم‌های رایانه‌ای کم‌کم در محیط اجرا و سرویس‌دهی به سبب مسئله‌های نگهداری و پشتیبانی کارایی خود را از دست داده و خطاهایی را در زمان اجرا در سیستم در پی خواهند داشت.

برای شناسایی خطاها در زمان اجرا و جلوگیری از شکست در سیستم از روش تحمل‌پذیری خطا بهره برده می‌شود. این روش به سیستم کمک می‌کند تا در هنگام رخداد خطا نیز به سرویس‌دهی خود ادامه دهد. در این روش برای رسیدن به قابلیت اطمینان بالا در سیستم از شگرد تکرار داده بهره برده شده است [۲][۳].

۱-۲- طرح مسئله

در سیستم‌های توزیع شده برای رسیدن به کارایی (دسترس‌پذیری بر روی داده) و قابلیت اطمینان بالا، شگرد تکرار داده به کار گرفته می‌شود [۲]. یکی از ویژگی‌های مهم این شگرد، یکسان بودن تکرارها از دیدگاه نرم‌افزاری می‌باشد. از سوی دیگر کمبودهای نرم‌افزاری رخداد خطاهای نرم‌افزاری در داده‌های تکرار را به همراه خواهند داشت [۴]. بنابراین می‌توان گفت اگر خطایی در یکی از تکرارها رخ دهد آن‌گاه احتمال رخداد همان خطا تحت آن شرایط در بقیه تکرارها نیز وجود خواهد داشت. حال اگر یک خطا به صورت تکراری در داده‌های تکرارشده رخ دهد، گسترش خطا در سیستم پدید می‌آید. به طور کلی رخداد هر خطا در سیستم هزینه تحمل‌پذیری خطا را در پی خواهد داشت [۳]. مسئله بنیادی رخداد پدیده گسترش خطا در سیستم می‌باشد که این پدیده هزینه تحمل‌پذیری خطای کل سیستم را به صورت فزاینده افزایش خواهد داد.

از این رو، مسئله بنیادی رخداد پدیده گسترش خطا در سیستم می‌باشد. در این پژوهش راه‌حلی به نام ارائه روشی برای جلوگیری از گسترش خطا در یک محیط توزیع‌شده، برای حل این مسئله ارائه شده است. در این روش خطاهای رخداده بر روی سیستم به‌وسیله کارشناس تشخیص خطا شناسایی می‌شود. پس از آن ویژگی‌های مرتبط با خطای درخواست‌های فرستاده شده در پایگاه‌داده ذخیره می‌شود. در گام پسین اگر دوباره همان درخواست فرستاده شود، کارشناس تشخیص خطا درخواست مورد نظر را پردازش نکرده، از رخداد دوباره خطا جلوگیری خواهد کرد.

۱-۳- هدف پژوهش

هدف از این پژوهش ارائه روشی برای جلوگیری از گسترش خطا در یک محیط توزیع شده در زمان اجرا می‌باشد. این روش در پایان هزینه تحمل‌پذیری‌خطای کل سیستم را با جلوگیری از تکرار دوباره خطاهای یکسان بر روی تکرارها کاهش خواهد داد.

۱-۴- محدوده پژوهش

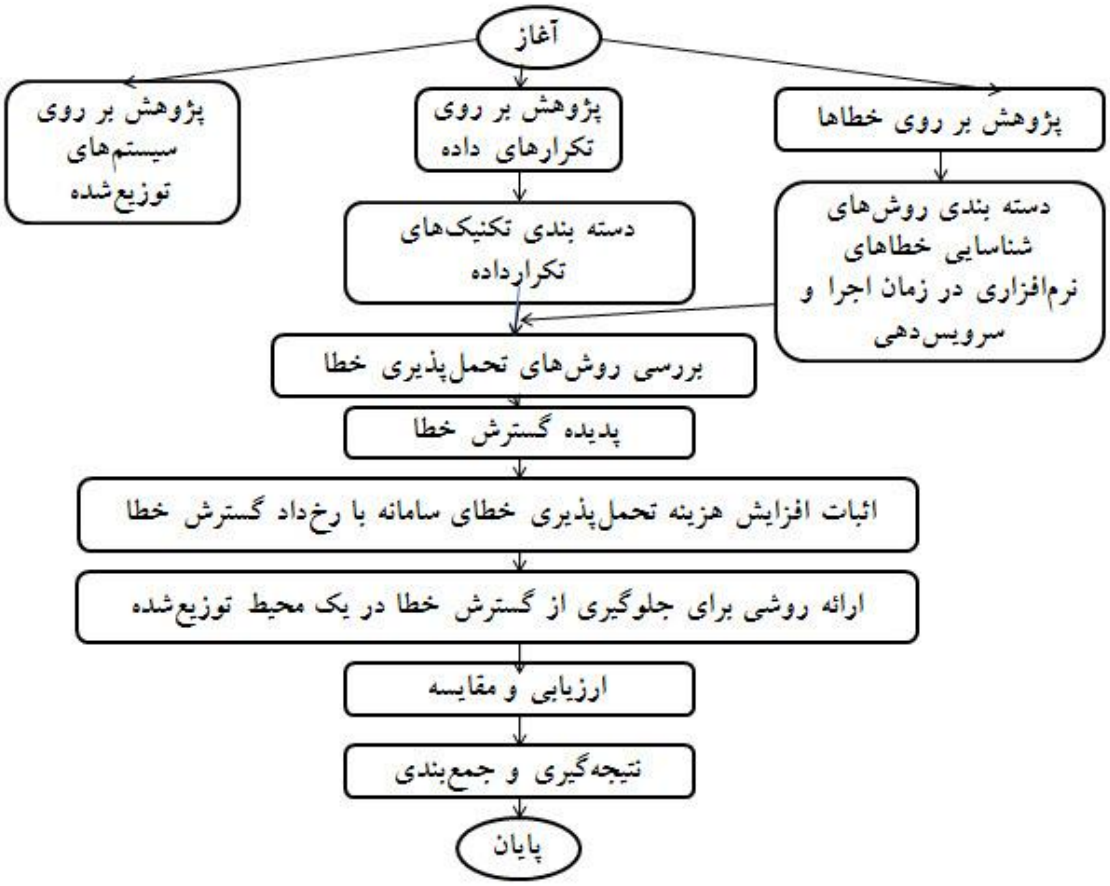
همان‌گونه که پیش از این گفته شد، برای افزایش قابلیت اطمینان سیستم توزیع شده از دو روش پیش‌گیری از خطا و تحمل‌پذیری خطا بهره گرفته شده است. روش پیش‌گیری از خطا در زمان پیش از اجرا و سرویس‌دهی و روش تحمل‌پذیری خطا در زمان اجرا و سرویس‌دهی کارهای خود را برای بالابردن قابلیت اطمینان سیستم‌ها انجام می‌دهند.

باید توجه داشت این پژوهش تنها به شناسایی خطاهای نرم‌افزاری در زمان اجرا و سرویس‌دهی مولفه‌های نرم-افزاری درون سیستم‌های توزیع شده پرداخته و خطاهای سخت‌افزاری و نرم‌افزاری پیش از زمان اجرا و سرویس-دهی را بررسی نمی‌کند. بنابراین با شناسایی خطاهای نرم‌افزاری در زمان اجرا از رخداد دوباره آن خطا در مولفه‌های نرم‌افزاری جلوگیری شده، قابلیت اطمینان سیستم افزایش خواهد یافت.

۱-۵- مراحل انجام پژوهش

برای پاسخ‌گویی به مسئله طرح شده در این پژوهش، از پژوهش‌های کتابخانه‌ای برای شناسایی دانسته‌های مورد نیاز بهره برده شده است. در این پژوهش به ترتیب بر روی سیستم‌های توزیع شده، تکرار داده و خطاها بررسی‌هایی انجام شده است. سپس شگردهای تکرار داده، روش‌های پیش‌گیری از خطا، تحمل‌پذیری خطا و شناسایی خطاهای نرم‌افزاری در زمان اجرا و سرویس‌دهی سیستم دسته بندی شده است.

پس از آن با بررسی روش‌های تحمل‌پذیری خطا پی به مشکل این روش برده شد. مشکل این روش هزینه تحمل‌پذیری خطا می‌باشد که در زمان اجرا پس از شناسایی خطا برای سیستم تحمیل می‌شود. سپس با پژوهش‌های انجام شده بر روی شگردهای تکرار داده، روش‌های شناسایی خطاهای نرم‌افزاری و هزینه تحمل‌پذیری خطا، پدیده‌ای به نام گسترش خطا ارائه شده است. این پدیده هزینه تحمل‌پذیری خطای سیستم را افزایش خواهد داد. برای جلوگیری از این پدیده روشی پیشنهاد شده است. در ادامه این روش بر روی سیستم توزیع شده تحمل‌پذیر خطا قرار داده شده و با سیستم توزیع شده تحمل‌پذیر خطا مقایسه و ارزیابی می‌شود. در پایان نیز به جمع‌بندی و نتیجه‌گیری‌های به‌دست آمده از این پژوهش پرداخته شده است. در شکل ۱-۱ این گام‌ها نشان داده شده است.



شکل۱-۱ نمایی از ساختار گام‌های انجام پایان‌نامه

۱-۶- ساختار پایان‌نامه

ساختار این پایان‌نامه به صورت زیر است:

در فصل دوم دانسته‌های پایه‌ای بهره برده شده در این پایان‌نامه مانند سیستم‌های توزیع شده و روش‌های شناسایی خطا شرح داده شده است. در فصل سوم، روش پیشنهادی به صورت فراگیر بیان شده و در فصل چهارم به ارزیابی روش پیشنهادی پرداخته و در پایان در فصل پنجم با نتیجه‌گیری و بیان کارهای آینده، این پژوهش به پایان رسیده است.

فصل دوم

دانسته‌های پایه‌ای و پیشینه پژوهش

۲-۱- پیش‌گفتار

از آنجا که دانسته‌ها و واژگان زیادی در این حیطه به کار برده شده، در این فصل به معرفی و توضیح‌های این واژگان و دانسته‌ها پرداخته خواهد شد. علاوه بر معرفی دانسته‌ها و واژگان در این فصل به بررسی روش‌های ارائه شده برای شناسایی و تحمل‌پذیری خطاها پرداخته و نقطه‌های ضعف و قوت این روش‌ها توضیح داده خواهند شد. در پایان نیز با جمع‌بندی نقطه‌های ضعف و قوت روش‌های موجود به تبیین ویژگی‌های روش مورد انتظار برای ارائه روشی برای جلوگیری از گسترش خطا در یک محیط توزیع‌شده پرداخته خواهد شد.

۲-۲- سیستم‌های توزیع شده

هر سیستم‌ای که بر روی مجموعه‌ای از ماشین‌ها که دارای حافظه اشتراکی نیستند، اجرا شده و برای کاربرها به گونه‌ای اجرا شود که گویا بر روی یک کامپیوتر می‌باشد، یک سیستم توزیع شده است. این تعریف یک تعریف بسیار کلی و بدون پشتوانه علمی است، به همین سبب دو تعریف رسمی در زیر ارائه شده است.

در کتاب سیستم‌های توزیع شده تعریفی به صورت زیر ارائه شده است:

به مجموعه‌ای از کامپیوترهای مستقل که برای کاربرها به عنوان یک سیستم واحد و به هم پیوسته نمایان می-شوند، سیستم‌های توزیع شده گفته می‌شود [۲].

در کتاب محاسبه‌های توزیع شده تعریفی به صورت زیر بیان شده است:

به یک مجموعه از موجودیت‌های مستقل از هم که با هم همکاری می‌کنند تا مسئله‌هایی را که به صورت فردی حل نمی‌شود را حل کنند، سیستم‌های توزیع شده گفته می‌شود [۵].

در این تعریف‌ها هیچ توضیحی در مورد نوع کامپیوترها و نوع ارتباط‌های میان آنها داده نشده است. ولی از دو جنبه بالا می‌توان نتیجه گرفت که یک همکاری میان کامپیوترهای مستقل امری ضروری است. در اینجا از پرداختن بیشتر به تعریف‌ها صرفنظر می‌شود [۲].

۲-۳- هدف‌های موجود در سیستم‌های توزیع شده

در بخش ۲-۲ تعریف‌ها و ویژگی‌هایی از سیستم‌های توزیع شده بیان شده است. بنابراین با توجه به موردهای گفته شده می‌توان نتیجه گرفت که سیستم‌های توزیع شده چهار هدف زیر را دنبال می‌کنند:

- دسترس‌پذیری به داده‌ها یا منبع‌ها
- شفافیت توزیع[۱]
- باز بودن[۲]
- توسعه‌پذیری[۳]

حال در زیر به بررسی و توضیح هر کدام از این هدف‌های بیان شده پرداخته می‌شود [۲]:

۲-۳-۱- دسترس‌پذیری به داده‌ها یا منبع‌ها

هدف اصلی سیستم‌های توزیع شده این است که کاربرها و برنامه‌های کاربرها بتوانند به راحتی به منبع‌های غیر محلی[۴] دسترسی داشته باشند و آنها را به روش کنترل شده و موثر به اشتراک بگذارند. منبع‌ها می‌توانند هر چیزی باشند. نمونه‌هایی از این منبع‌ها عبارتند از چاپگرها، کامپیوترها، جاهای ذخیره‌سازی داده‌ها، فایل‌ها، صفحه‌های وب و شبکه‌ها. علت‌های زیادی برای اشتراک منبع‌ها وجود دارد. یک دلیل روشن، صرفه‌جویی اقتصادی است. برای

---

[1] Distribution transparency
[2] Openness
[3] Scalability
[4] Remote resource