

ارائه چارچوب آزمون خودکار عملکرد سیستم‌های مبتنی بر وب و تحلیل نتایج آن با استفاده از معیارهای کمی

بهشید بهکمال محسن کاهانی سهیلا دهقانزاده

^۱دانشکده فنی و مهندسی، دانشگاه فردوسی مشهد، مشهد، ایران

چکیده

با توجه به رشد روزافزون کاربردهای مختلف نرم‌افزارهای مبتنی بر وب، موضوع کیفیت این سیستم‌ها از اهمیت بالایی برخوردار است و به دلیل تنوع و ناهمگونی در محیط اجرای آنها، روش‌های رایج برای آزمون نرم‌افزارهای متداول برای تست و ارزیابی سیستم‌های تحت وب کارآمد نمی‌باشند. هدف اصلی این مقاله ارائه چارچوبی برای آزمون است که با ویژگی‌های محیطی و اجرایی این سیستم‌ها سازش و تطابق داشته باشد. از این رو ابتدا با مطالعه سیستم‌های مبتنی بر وب و بررسی نیازمندی‌های آزمون این سیستم‌ها چارچوبی برای آزمون خودکار عملکرد سیستم‌های مبتنی بر وب و تحلیل نتایج آن ارائه می‌شود سپس بر اساس معیارهای ارزیابی کیفیت نرم‌افزار، سه معیار کمی به منظور ارزیابی خروجی‌های چارچوب پیشنهاد خواهد شد. در پایان چارچوب پیشنهادی برای آزمون خودکار سه نمونه سیستم مبتنی بر وب با محیط‌های اجرای متفاوت بطور تجربی بکار گرفته شده و نتایج آزمون با استفاده از معیارهای کمی مورد ارزیابی قرار خواهد گرفت.

کلمات کلیدی: آزمون نرم‌افزار، معیارهای کمی، سیستم‌های مبتنی بر وب، خودکارسازی آزمون.

۱- مقدمه

پاسخگویی به کاربران مختلف را فراهم نمایند. همچنین، در سیستم‌های تحت وب، بین برنامه سمت سرور و برنامه سمت کلاینت، ارتباط تنگاتنگی وجود ندارد و اگر برنامه سمت سرور دچار خرابی شود و اجرای آن با مشکل مواجه شود، کاربر سیستم اطلاعات مناسبی برای تشخیص علت خرابی ندارد. این کمبود اطلاعات، عدم آگاهی از جزئیات دقیق وضعیت سیستم، و عدم امکان دستکاری و انجام آزمون و خطا بر روی آن، طبیعتاً بر عملیات تست سیستم تأثیر می‌گذارد. با توجه به ویژگی‌ها و پیچیدگی‌های خاصی که اشاره شد، روش‌ها، ابزارها و مدل‌های رایج برای آزمون نرم‌افزارهای متداول، معمولاً برای تست و ارزیابی سیستم‌های تحت وب، کارآمد نمی‌باشند. برخی از این روش‌ها نیازمند تغییر و تطبیق با محیط وب می‌باشند و برخی نیز بکلی قابل استفاده نمی‌باشند. همچنین برای برخی از موارد آزمون، روش‌ها و مدل‌های جدید که مخصوص سیستم‌های تحت باشند، مورد نیاز است [۲]، [۱] و خودکارسازی فرآیند آزمون راه حل مناسبی برای کاهش هزینه‌های این آزمون می‌باشد [۳].

در حال حاضر، تفاوت بین دو نرم‌افزار را توانایی نرم‌افزارها در برآورده ساختن ویژگی‌های کیفی مورد انتظار که همان کیفیت نرم‌افزار است، تعیین می‌کند. آزمون نرم‌افزار بعنوان یک راه حل مهم در زمینه سنجش کیفیت نرم‌افزار ارائه شده که باید در مراحل مختلف چرخه عمر نرم‌افزار انجام شود. نرم‌افزارهای مبتنی بر وب نیز از این قاعده مستثنی نیستند، ولی به دلیل آنکه مدل رایج برای توسعه این سیستم‌ها، روش توسعه تکاملی تدریجی است، این نرم‌افزارها همواره در حال تغییر و تکامل هستند و زمان بسیار کوتاه ارائه به بازار، موجب می‌شود تا توسعه‌دهندگان سیستم، فرصت کافی برای انجام آزمون‌های لازم را نداشته باشند. از سوی دیگر، محیط اجرای سیستم‌های تحت وب از پویایی و تنوع بسیار بالایی برخوردار است. در نتیجه طراحان سیستم‌های تحت وب، با حجم بالایی از ناهمگونی در محیط اجرای سیستم مواجه هستند که باید به شکلی با مدیریت این مساله، امکان

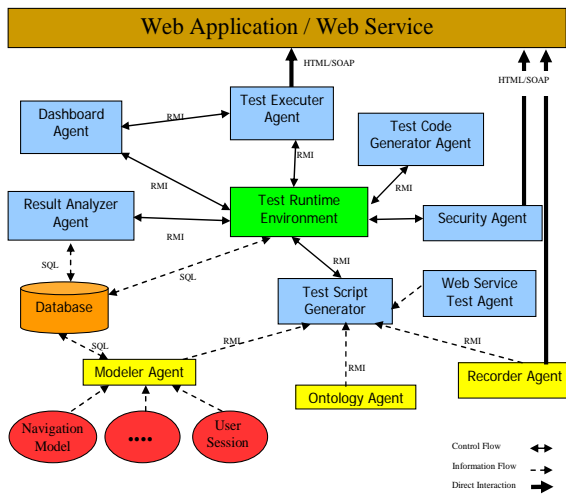
سیستم‌های مبتنی بر عامل در کاربردهای مختلفی که در یک بستر توزیع شده نظیر اینترنت اجرا می‌شوند، مورد استفاده قرار گرفته و با توجه به مزایایی که فراهم می‌کنند، همواره مورد توجه بوده‌اند.

برای پیاده‌سازی یک سیستم مبتنی بر عامل، مناسب‌ترین روش، استفاده از میان‌افزارهایی^۱ می‌باشد که مخصوص این کار طراحی شده‌اند. با استفاده از سرویس‌های پایه‌ای که این میان‌افزارها ارائه می‌کنند، پیاده‌سازی سیستم مورد نظر، از سرعت و کیفیت بالاتری برخوردار خواهد بود. از دیگر نکات مثبت استفاده از این میان‌افزارها، انطباق با استانداردهای مختلف نرم‌افزاری است که در این زمینه مطرح می‌باشد.

در این مقاله از یک میان‌افزار کد منبع باز مطرح در زمینه سیستم‌های مبتنی بر عامل که مبتنی بر زبان برنامه‌نویسی قدرتمند جاوا می‌باشد استفاده شده است.^۲

۲-۱- مدل کلی چارچوب

چارچوب بصورت یک سیستم چند عامله معرفی می‌شود که عامل‌ها در آن برای جداسازی فعالیت‌های مختلف و پیواسازی سیستم بکار می‌روند. عامل‌های مختلف در سرتاسر شبکه توزیع شده و می‌توانند در دسترس باشند یا شبکه را ترک کنند. فرایند آزمون به کمک این عامل‌ها انجام می‌شود و یک بخش مرکزی، مدیریت سیستم را بر عهده دارد. در شکل ۲ مدل کلی چارچوب ارائه شده است.



شکل ۲- مدل کلی چارچوب

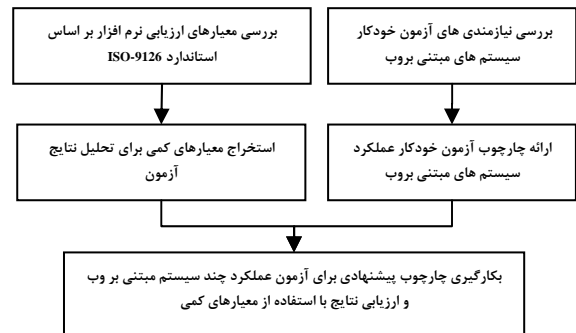
۲-۲- عامل‌های اصلی سیستم

محیط اجرای آزمون^۳، بخش مرکزی سیستم است که با دیگر بخش‌های سیستم ارتباط برقرار کرده تا بتواند وظیفه مدیریت تنظیم و اجرای فعالیت‌های مختلف آزمون را بدرستی انجام دهد. همچنین، این بخش وظیفه فراهم نمودن یک واسط کاربری مناسب برای ایجاد و اجرای آزمون‌های کاربران را برعهده دارد. در ابتدا کاربر سیستم، تمامی تنظیمات مربوط به تست را با استفاده از رابط کاربری عامل محیط اجرای آزمون انجام می‌دهد، سپس عامل محیط اجرای آزمون با توجه به میزبان‌های موجود در سیستم و تعداد عامل‌ها روی هر میزبان، اسکریپت تست مورد نیاز و تنظیمات لازم برای اجرای تست را به میزبان‌ها می‌فرستد.

عامل مولد کد آزمون^۴ اسکریپت آزمون را به عنوان ورودی گرفته و با کامپایل آن کد جاوای متناظر با آن را تولید می‌کند. باید توجه داشت که دو مفهوم مشابه

ساختار مقاله در شکل ۱ آورده شده است، در مقدمه این مقاله با مروری بر ویژگی‌های سیستم‌های مبتنی بر وب، ضرورت آزمون خودکار این سیستم‌ها بررسی شد. در بخش دوم، یک چارچوب چندعامله با استفاده از ابزارهای کد منبع باز معرفی خواهد شد که هدف آن، آزمون خودکار عملکرد سیستم‌های مبتنی بر وب می‌باشد. برای ارزیابی نتایج آزمون لازم است تا معیارهای کمی تعریف شود که در بخش سوم، با بررسی اجمالی معیارهای ارزیابی نرم‌افزار، معیارهای مناسب برای تحلیل نتایج آزمون استخراج خواهد شد.

در پایان، بمنظور بکارگیری عملی چارچوب پیشنهادی، نتایج آزمون خودکار عملکرد سه نمونه سیستم مبتنی بر وب ارائه شده و این نتایج با استفاده از معیارهای کمی استخراج شده، بصورت کمی مورد مقایسه و ارزیابی قرار خواهد گرفت.



شکل ۱- ساختار کلی مقاله

در بخش بعد چارچوب پیشنهادی برای آزمون خودکار عملکرد سیستم‌های مبتنی بر وب ارائه خواهد شد.

۲- ارائه چارچوب پیشنهادی

برای تست سیستم‌های مبتنی بر وب که ذاتاً سیستم‌هایی پیچیده، چند لایه، توزیع شده و پویا می‌باشند، به چارچوبی نیاز داریم که با ویژگی‌های محیطی و اجرایی این سیستم‌ها، سازش و تطابق داشته باشد. طبیعتاً برای این منظور، یک چارچوب توزیع شده، از توانایی‌های بالاتری برخوردار است. با توجه به پیچیدگی‌های قابل انتظار در طراحی چارچوب مورد نظر، استفاده از مدل طراحی مبتنی بر عامل، امکان توزیع و تقسیم کارها و وظایف مختلف چارچوب را فراهم کرده و مزایایی نظیر افزایش انعطاف‌پذیری، مقیاس‌پذیری، و همچنین پویایی سیستم را به دنبال خواهد داشت.

وظایف مختلف سیستم بین عامل‌های مختلف تقسیم شده و این عامل‌ها ضمن تعامل با یکدیگر، اهداف سیستم را برآورده می‌سازند. با طراحی مناسب نقاط تماس عامل‌ها، این امکان وجود دارد که در زمان اجرای چارچوب، عامل‌های جدیدی به آن اضافه شده و یا عامل‌هایی از سیستم حذف شوند. بطور خلاصه، اصولی که در طراحی چارچوب پیشنهادی مدنظر قرار گرفته، بشرح زیر می‌باشد:

- مبتنی بر عامل
- اجرای تست‌ها براساس روش جعبه سیاه
- تهیه موارد آزمون براساس روش‌های جعبه سفید
- استفاده از ابزارهای کد منبع باز موجود (تا حد ممکن)
- پشتیبانی از پروتکل‌ها و تکنیک‌های جدید
- مستقل از سکو بودن

- قابلیت فهم
- عدم همپوشانی بین سطوح مدل (دارای ساختار یک به چند)
- دسته‌بندی جامع و کامل ویژگی‌های کیفیت نرم‌افزار همراه با تعریف واضح و دقیق آنها
- داشتن معیارهای کمی برای اندازه‌گیری ویژگی‌های کیفی

به دلیل ویژگی‌های خاص سیستم‌های مبتنی بر وب که پیش تر به آنها اشاره شد، معمولاً معیارهای متداول برای آزمون سایر نرم‌افزارها، برای آزمون سیستم‌های تحت وب، کافی و کارآمد نمی‌باشند. برخی از این معیارها نیازمند تغییر و تطبیق با محیط وب می‌باشند و برخی نیز بکلی قابل استفاده نمی‌باشند. در این مقاله، به منظور انتخاب معیارهای کمی، فاکتورهای زیر مدنظر قرار گرفته است:

۱. معیارهای کمی مدل استاندارد بر دو دسته‌اند: برخی از آنها برای ارزیابی کیفیت در مراحل مختلف توسعه نرم‌افزار مورد استفاده قرار می‌گیرد و دسته دوم برای ارزیابی محصول نهایی. با توجه به اینکه هدف این چارچوب، آزمون محصول نرم‌افزاری است، از معیارهایی که برای آزمون محصول نهایی ارائه شده، استفاده شده است.
۲. معیارهای کمی ارائه شده برای استفاده فرد آزمون‌گر تعریف شده است و همه معیارها برای بکارگیری توسط عامل نرم‌افزاری قابل استفاده نیست. از آنجایی که قرار است آزمون نرم‌افزار بطور خودکار انجام شود، از معیارهایی که قابلیت ارزیابی توسط عامل نرم‌افزاری دارند، استفاده شده است.
۳. معیارهای ارائه شده در استاندارد برای استفاده ذینفعان مختلف نرم‌افزار طراحی شده است. این ذینفعان عبارتند از توسعه دهنده^۹، نگهدارنده^۸، آزمون‌گر^{۱۰}، طراح واسط کاربر^{۱۱}، تأییدکننده کیفیت^{۱۲}، تأمین کننده^{۱۳} و کاربر^{۱۴}. در این تحقیق از معیارهایی استفاده شده که توسط آزمون‌گر و تاحدی توسعه دهنده و کاربر بکار گرفته می‌شود.
۴. از آنجایی که در این چارچوب، نرم‌افزار از جنبه سطح کیفی عملکرد آن مورد آزمون و ارزیابی قرار می‌گیرد، لذا فقط از معیارهایی که برای اندازه‌گیری ویژگی کیفی عملیاتی بودن^{۱۵} در مدل استاندارد ارائه شده، استفاده شده است.
۵. معیارهای مدل استاندارد حالت کلی و عمومی دارد و نمی‌توان از آنها برای ارزیابی انواع نرم‌افزارها استفاده نمود. لذا از بین آنها، معیارهایی که پس از اصلاح و تغییر برای ارزیابی نرم‌افزارهای مبتنی بر وب مناسب هستند، استخراج شده است.

براساس عوامل پنج گانه فوق، از بین معیارهای مدل استاندارد سه معیار کمی کفایت عملکردی^{۱۶}، فرکانس خطا^{۱۷} و زمان پاسخگویی^{۱۸} که برای آزمون عملکرد نرم‌افزارهای مبتنی بر وب قابل استفاده هستند، استخراج شده است.

• معیار اول: کفایت عملکردی

این معیار برای ارزیابی کفایت عملکردی نرم‌افزار مورد آزمون است و براساس فرمول ۱ اندازه‌گیری می‌شود:

(۱)

$$1 - \frac{\text{number of failed steps executed by all threads on all hosts}}{\text{number of total steps executed by all threads on all hosts}}$$

روی میزبان‌های تعریف شده توسط کاربر آزمون، تعدادی عامل چندین نمونه تست را بصورت همزمان اجرا می‌کنند. هر سناریوی آزمون که شامل چندین مرحله است، در قالب یک نخ^{۱۹} به سرور وصل می‌شود. چنانچه هر مرحله تست از یک نخ عامل، روی یک میزبان شکست بخورد، نشان‌دهنده عدم کفایت عملکردی نرم‌افزار در پاسخگویی به درخواست‌های همزمان است. بنابراین معیار فوق هر چه

در چارچوب وجود دارد: اسکرپیت آزمون و کد آزمون. اسکرپیت آزمون یک نمایش متنی از فرایند آزمون است که با استفاده از سلنیوم و یا یکی از عامل‌های مدل کننده، عامل تست وب سرویس، عامل تست امنیت و یا عامل ایجاد کننده اسکرپیت تست ایجاد شده است، در حالی که کد آزمون کد جاوای اسکرپیت آزمون است که باید برای اجرا ابتدا توسط عامل زمان اجرا کامپایل و نسخه باینری آن به عامل‌های اجرای آزمون فرستاده می‌شود.

عامل اجرای آزمون^۵ وظیفه اجرای آزمون را روی برنامه‌ها یا سرویس‌های وب برعهده دارد. محیط اجرا نسخه اجرایی تولید شده توسط خود را برای اجرا به عامل اجرا تحویل می‌دهد. عامل اجرا آزمون را انجام داده و نتایج آزمون را در فایل ذخیره می‌کند. پس از خاتمه اجرا، عامل اجرا فایل نتیجه را برای محیط اجرا می‌فرستد تا در پایگاه داده سیستم ذخیره کند و در حین اجرای تست نیز هر چند وقت یکبار نتایج تست را برای عامل داشبورد می‌فرستد تا داشبورد را به هنگام کند. (فرکانس بهنگام‌سازی داشبورد ابتدا توسط کاربر تنظیم می‌شود).

عامل داشبورد^۶ در حین اجرای آزمون نتایج تست را به کاربر نشان می‌دهد. این نتایج را عامل اجراکننده تست هر چند وقت یکبار در حین اجرای آزمون برای عامل داشبورد می‌فرستد تا داشبورد یک نمایش بلادرنگ از وضعیت اجرای آزمون و نتایج آن ارائه دهد. این اطلاعات شامل مراحل می‌باشد که عامل اجرا کننده در حال اجرای آن است، اگر یک مرحله از تست شکست بخورد، نمودار میله‌ای عامل داشبورد برای مرحله جاری یک میله هاشور زده شده اضافه می‌کند و اگر این مرحله با موفقیت پاس شود، نمودار میله‌ای عامل داشبورد برای مرحله جاری یک میله آبی رنگ اضافه می‌کند.

محیط آزمون، نتایج آزمون را از عامل اجرا دریافت و به عامل تحلیل گر نتایج^۷ می‌فرستد تا تحلیلی براساس آنچه که کاربر مشخص کرده روی آنها انجام شود. عامل تحلیل گر می‌تواند نتایج آزمون را در قالب گزارشاتی با فرمت‌ها، گراف‌ها و جداول مختلف ارائه دهد. این عامل از معیارهای نرم‌افزاری برای ارزیابی استفاده می‌کند که در بخش بعدی به آن پرداخته خواهد شد.

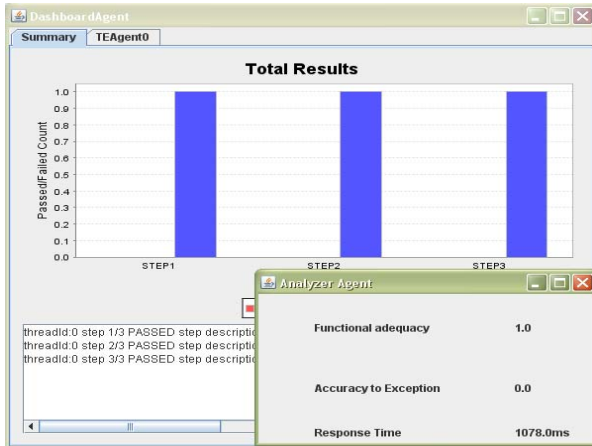
۳- استخراج معیارهای کمی برای تحلیل نتایج آزمون

معیارهای نرم‌افزار برای برای کنترل کیفیت نرم‌افزار، ارزیابی سودآوری سیستم، کنترل پروژه و تخمین نیازمندی‌ها مورد استفاده قرار می‌گیرند. طبقه‌بندی‌های متعددی برای معیارهای نرم‌افزار از دیدگاه افراد مختلف و با توجه به کاربرد آنها در حوزه‌های نرم‌افزارهای مختلف ارائه شده است [۴]، [۵]، [۶]، [۷]، [۸]، [۹] و [۱۰] که بطور خلاصه می‌توان به این موارد اشاره کرد: دسته‌بندی براساس کیفی یا کمی بودن، ایستا یا پویا بودن، تابعی یا شیئی گرابی بودن طراحی، داخلی یا خارجی بودن. از آنجایی که هدف چارچوب پیشنهادی، آزمون عملکرد نرم‌افزار می‌باشد، باید معیارهایی برای ارزیابی عملکرد ارائه شود.

استاندارد ISO/IEC 9126 که در سال ۲۰۰۱ توسط سازمان جهانی استاندارد برای ارزیابی محصول نرم‌افزار و ویژگی‌های کیفیت و راهنمایی‌های لازم برای استفاده از آنها ارائه شده است [۱۱]، یک مدل کیفیت را برای ارزیابی محصول نرم‌افزاری توصیف می‌کند. معیارهای تعریف شده در این مدل، هم برای ارزیابی نیازهای عملکردی و هم نیازهای غیر عملکردی کاربران و خریداران استفاده می‌شود.

با توجه به اینکه این استاندارد در مقایسه با سایر مدل‌های کیفیت نرم‌افزار [۱۲]، [۱۳] برای آزمون خودکار نرم‌افزار مناسب‌تر است، در این مقاله از معیارهای این مدل برای آزمون عملکرد نرم‌افزارهای مبتنی بر وب استفاده شده است. از مهمترین دلایل انتخاب این مدل می‌توان به موارد زیر اشاره کرد:

تمامی مراحل موفقیت‌آمیز باشد و تعداد مراحل شکست خورده برابر صفر و اندازه معیار کفایت عملکردی برابر ۱ و فرکانس رخداد خطا برابر صفر و زمان پاسخگویی نیز خیلی کم باشد که نتیجه آزمون، تایید کننده این انتظار است.



شکل ۳- نتایج آزمون عملکرد خودکار میزبان محلی

همانگونه که در شکل ۳ آمده است، معیار کفایت عملکردی برابر ۱ شده است یعنی در تمامی نمونه‌های اجرا شده تمامی مراحل با موفقیت اجرا شده و معیار فرکانس رخداد خطا نیز برابر ۰ است که نشان‌دهنده عدم رخداد خطا در حین آزمون وب سایت است. همچنین زمان پاسخگویی حدود ۱ ثانیه است.

۴-۲- ارزیابی نتایج آزمون عملکرد سایت دانشکده ریاضی

در یک نمونه تست روی سایت دانشکده ریاضی دانشگاه فردوسی مشهد، تعداد مراحل آزمون و مراحل شکست خورده و موفق در شکل ۴ نشان داده شده است. این آزمون نیز شامل ۹ مرحله بوده است که با توجه به نمودار میله‌ای زیر مراحل ۳ و ۵ و ۹ که بصورت هاشور زده نشان داده شده‌اند، شکست خورده و بقیه مراحل با موفقیت انجام شده است.

کفایت عملکردی نرم‌افزار در اینجا روی شبکه داخلی بررسی می‌شود و با توجه به کمبود ترافیک انتظار می‌رود تعداد مراحل شکست خورده به نسبت کم باشد و معیار کفایت عملکردی تا حد زیادی به واقع نشان‌دهنده کفایت نرم‌افزار است و اندازه آن نیز به یک نزدیک می‌باشد که نشان‌دهنده تعداد کم مراحل شکست خورده است و فرکانس رخداد خطا نیز به نسبت کم است و زمان پاسخگویی نیز تقریباً زیاد است. همانگونه که در شکل ۴ نشان داده شده است، معیار کفایت عملکردی این سایت، پس از ۱۰ بار اجرای آزمون برابر ۰.۶ و فرکانس رخداد خطا ۰.۳ و زمان پاسخگویی آن حدود ۷ ثانیه است.

۴-۳- ارزیابی نتایج آزمون عملکرد وب سایت Alibaba

به عنوان یک نمونه سایت خارجی نیز سناریوی ورود به سایت <http://www.alibaba.com/> با آزمون بار^{۲۱} برای ده کاربر همزمان اجرا شده و نتایج آن در شکل ۵ آمده است. از آنجایی که در این آزمون بستر ارتباطی، شبکه اینترنت بوده و اتصال از طریق پروکسی انجام می‌شود، بدلیل ترافیک شبکه، کفایت عملکردی به خوبی نشان‌دهنده کفایت واقعی نرم‌افزار نیست.

به یک نزدیک تر باشد نشان‌دهنده این است که سیستم از نظر عملکرد از کیفیت بالاتری برخوردار است.

• معیار دوم: فرکانس خطا

این معیار برای ارزیابی فرکانس رخداد خطا در سیستم است و براساس فرمول ۲ اندازه‌گیری می‌شود:

(۲)

$$\frac{\text{number of failed steps}}{\text{time of total testScript Execution}}$$

تعداد مراحل شکست خورده نسبت به زمان اجرای لازم برای اجرای کل مراحل، فرکانس رخداد خطا را نشان می‌دهد. هر چه فرکانس رخداد خطا بیشتر باشد، یعنی قابلیت سرور برای پاسخگویی کمتر است.

• معیار سوم: زمان پاسخگویی

این معیار زمان پاسخگویی متوسط مراحل را نشان می‌دهد و معیاری از زمان پاسخگویی سرور می‌باشد و با استفاده از فرمول ۳ محاسبه می‌شود:

(۳)

$$\frac{\sum_{\text{hosts}} \sum_{\text{thread}} \sum_{\text{steps}} (\text{finishtime} - \text{starttime})}{\text{hosts} * \text{thread} * \text{steps}}$$

زمان شروع و خاتمه هر مرحله آزمون در هر نخ عامل به تفکیک میزبان‌ها در پایگاه داده سیستم ذخیره می‌شود و برای محاسبه متوسط زمان پاسخگویی، تفاضل زمان خاتمه هر مرحله را از زمان شروع همان مرحله محاسبه کرده و سپس همه تفاضل‌ها را با هم جمع کرده و بر تعداد مراحل تقسیم می‌کنیم. بدین ترتیب زمان متوسط پاسخگویی متوسط سیستم بدست می‌آید.

۴-۴- ارزیابی خودکار سه سیستم مبتنی بر وب با استفاده از چارچوب پیشنهادی

در این بخش با استفاده از معیارهای پیشنهادی، نتایج آزمون عملکرد خودکار سه سیستم مبتنی بر وب تحلیل و ارزیابی خواهد شد. برای ارزیابی چارچوب، سه نوع محیط انتخاب شده است، سیستم اول میزبان محلی^{۲۰} است که آزمون در محیطی کاملاً محلی و فاقد هر گونه مانع شبکه و ترافیک انجام می‌شود. سیستم دوم سایت یکی از دانشکده‌های دانشگاه فردوسی است که نرم‌افزار روی بستر شبکه داخلی دانشگاه اجرا می‌شود، و سیستم سوم که یک وب سایت اینترنتی با ترافیک بالاست. با انتخاب این سه سیستم، چارچوب برای ارزیابی نرم‌افزارهای مبتنی بر وب روی بسترهای متفاوت میزبان محلی، شبکه داخلی و اینترنت مورد استفاده قرار می‌گیرد.

۴-۱- ارزیابی نتایج آزمون عملکرد میزبان محلی

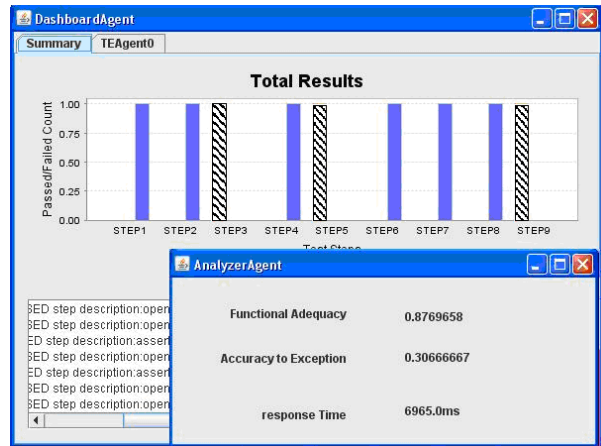
نمونه اول، یک میزبان محلی است که نتایج آزمون و مقادیر معیارهای حاصل از آزمون به همراه داشبورد سیستم، که نشان‌دهنده تعداد مراحل آزمون و مراحل دارای خطا و بدون خطاست، در شکل ۳ مشاهده می‌شود. این آزمون شامل ۳ مرحله بوده که با توجه به نمودار میله‌ای زیر، هر سه مرحله آزمون با موفقیت انجام شده است. بعبارت دیگر به دلیل نبود موانع شبکه و مسیریابی انتظار می‌رود،

سپاسگزاری

این پژوهش با حمایت پژوهشگاه تحقیقات ارتباطات و فناوری اطلاعات انجام گرفته است که بدین وسیله از این مرکز قدردانی می‌گردد. همچنین از اعضای محترم آزمایشگاه فناوری وب دانشگاه فردوسی مشهد^{۲۲} که در طراحی و پیاده‌سازی این چارچوب همکاری نموده‌اند، تشکر و قدردانی می‌گردد.

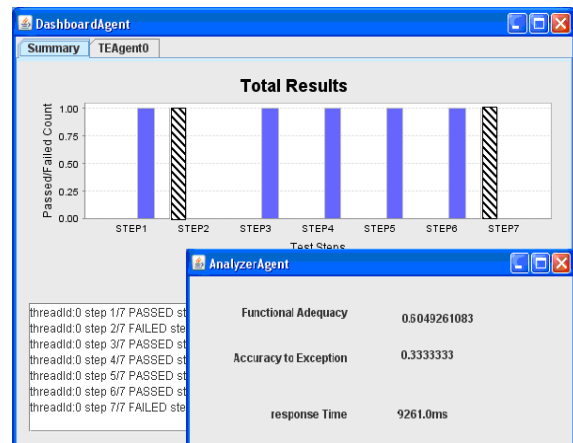
منابع

- [1] N. Mansour, and M. Hourri, "Testing Web Applications," *Information and Software Technology*, vol. 48, pp. 31-42, 2006.
- [2] A. Giuseppe, D. Lucca, and A. R. Fasolino, "Testing Web-based applications: The State of the Art and Future Trends," *Information and Software Technology*, vol. 48, pp. 1172-1186, 2006.
- [3] E. Hieatt, and R. Mee, "Going faster: Testing the Web Application," *IEEE Software*, vol. 19, pp. 60-65, 2003.
- [4] C. Calero, J. Ruiz, and M. Piattini, "Classifying web metrics using the web quality model," *Online Information Review Journal*, vol. 29, no. 3, pp. 645-661, 2005.
- [5] I. B. Hong, "A survey of web site success metrics used by Internet-dependent organizations in Korea," *Internet Research*, vol. 17, no. 3, pp. 272 - 290, 2007.
- [6] R. Pressman, *Software Engineering, a Practitioner's Approach*, McGrawHill, seventh Edition, 2010.
- [7] D. Dhyani, N. W. Keong, and S. Bhowmicka, "Survey of Web Metrics," *ACM Computing Surveys*, vol. 2, no. 3, pp. 1-42, 2002.
- [8] M. E. Wakil, A. Bastawisi, M. Boshra, and A. Fahmy, "Object-Oriented Design-Metrics - A taxonomy," *Third International Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Applications (CSITeA'04)*, 2004.
- [9] M. Piattini, and M. Serrano, "Classifying Web Metrics," *First Intl Workshop on Software Audit and Metrics*, pp. 22-37, 2004.
- [10] E. Mendes, S. Counsell, and N. Mosley, "Towards a Taxonomy of Hypermedia and Web Application Size Metrics," *Proceedings of the ICWE'05 Conference*, Springer LNCS 3579, pp. 110-123, 2005.
- [11] ISO/IEC 9126-1, *Software engineering - Product quality-Part 1: Quality Model*, First edition, 2001.
- [12] A. Moraga, C.I Calero, and M. Piattini, "Comparing different quality models for portals," *Online Information Review Journal*, vol. 30, no. 5, 2006.



شکل ۴- نتایج آزمون عملکرد خودکار وب سایت دانشکده ریاضی دانشگاه فردوسی

بنابراین انتظار می‌رود این معیار بسیار کم باشد و بدلیل مشابه، فرکانس بالای رخداد خطا و زمان پاسخگویی زیاد پیش‌بینی می‌شود.



شکل ۵- نتایج آزمون عملکرد خودکار وب سایت Alibaba

نتایجی که در شکل ۵ آمده است، تاییدکننده پیش‌بینی است و نشان می‌دهد معیار کفایت عملکردی این وب سایت برابر ۰.۸ و فرکانس رخداد خطا برابر ۰.۳ می‌باشد. مدت زمان پاسخگویی وب سایت نیز حدود ۹ ثانیه اندازه‌گیری شده است. این آزمون نیز شامل ۷ مرحله بوه است که با توجه به نمودار میله‌ای فوق، مراحل ۲ و ۷ شکست خورده‌اند و ۵ مرحله دیگر همگی موفقیت‌آمیز بوده است.

۵- نتیجه‌گیری

هدف این مقاله، ارائه چارچوبی برای آزمون خودکار عملکرد سیستم‌های مبتنی بر وب و ارزیابی نتایج این آزمون با استفاده از معیارهای کمی نرم‌افزاری بوده است. برای این منظور، ابتدا با مروری بر ویژگی‌های سیستم‌های مبتنی بر وب، نیازمندی‌های آزمون خودکار این سیستم‌ها، بررسی و سپس چارچوبی برای آزمون خودکار عملکرد این سیستم‌ها ارائه شد. سپس، براساس معیارهای ارزیابی نرم‌افزار، معیارهای مناسب برای تحلیل نتایج آزمون عملکرد استخراج شده و در پایان، با استفاده از معیارهای پیشنهادی، نتایج آزمون عملکرد سه نمونه سیستم مبتنی بر وب مورد ارزیابی قرار گرفت.

- 12 Software Quality Assurance (SQA)
 13 Supplier
 14 User
 15 Functionality
 16 Functional Adequacy
 17 Accuracy to Expectation
 18 Response Time
 19 Thread
 20 Local host
 21 Load Test
 22 <http://wtlab.um.ac.ir>

[13] S. H. Kan, *Metrics and Models in Software Quality Engineering*, Addison-Wesley Professional, 2009.



بهشید بهکمال دانشجوی دکتری مهندسی کامپیوتر دانشگاه فردوسی مشهد می‌باشد. وی دارای مدرک کارشناسی مهندسی کامپیوتر از دانشگاه فردوسی مشهد و مدرک کارشناسی ارشد مهندسی فناوری اطلاعات از دانشگاه صنعتی امیرکبیر می‌باشد. او از سال ۱۳۸۷ مشغول به تحصیل در مقطع دکتری است و در حال حاضر فعالیت پژوهشی وی متمرکز بر کیفیت داده، وب معنایی و داده‌های پیوندی می‌باشد. آدرس پست‌الکترونیکی ایشان عبارت است از:

behkamal@stu-mail.um.ac.ir



محسن کاهانی دانشیار گروه مهندسی کامپیوتر دانشگاه فردوسی مشهد می‌باشد. ایشان دکتری خود را در زمینه مهندسی کامپیوتر در سال ۱۳۷۷ از دانشگاه ولونگونگ استرالیا اخذ نمودند و از همان سال بعنوان عضو هیئت علمی در دانشگاه فردوسی مشهد مشغول به کار شدند. زمینه‌های تحقیقاتی ایشان مهندسی وب، وب معنایی، بازیابی اطلاعات و مهندسی نرم‌افزار می‌باشد. آدرس پست‌الکترونیکی ایشان عبارت است از:

kahani@um.ac.ir



سهیلا دهقانزاده مدرک کارشناسی و کارشناسی ارشد خود را در رشته مهندسی کامپیوتر از دانشگاه فردوسی مشهد به ترتیب در سال‌های ۱۳۸۷ و ۱۳۹۰ اخذ نموده است. او در حال حاضر دانشجوی دکتری در دانشگاه ملی ایرلند می‌باشد. نکته مرکزی در تحقیقات جاری او پیرامون کاوش وب، تحلیل شبکه‌های اجتماعی و بازیابی اطلاعات در زمینه وب معنایی است. آدرس پست‌الکترونیکی ایشان عبارت است از:

soheila.dehghanzadeh@stu-mail.um.ac.ir

اطلاعات بررسی مقاله:

تاریخ ارسال: ۸۸/۷/۷

تاریخ اصلاح: ۹۱/۶/۱۲

تاریخ قبول شدن: ۹۱/۸/۱۵

نویسنده مرتبط: بهشید بهکمال، دانشکده فنی و مهندسی، دانشگاه فردوسی مشهد، مشهد، ایران.

- 1 Middleware
 2 JADE
 3 Test Runtime Environment(TRE)
 4 Test Code Generator Agent (TCGA)
 5 Test Executer Agent (TEA)
 6 Dashboard Agent (DA)
 7 Result Analyzer Agent (RAA)
 8 Developer
 9 Maintainer
 10 Tester
 11 Human Interface Designer