

روشی به منظور ارزیابی قابلیت اطمینان سبک‌های چندریختی معماری نرم‌افزار

سپیده عظیمی*^۱، گلناز آقایی قزوینی^۲

دانشجویی کارشناسی ارشد، مهندسی نرم‌افزار دانشگاه آزاد اسلامی واحد دانشگاه آزاد

اسلامی (خوراسگان)، ایران

Sepidehazimi@yahoo.com

دکتری، عضو هیات علمی دانشگاه آزاد اسلامی واحد دولت آباد

golnazghazvini@yahoo.com

چکیده

طراحی معماری نرم‌افزار نقش بسیار مهمی در چرخه حیات نرم‌افزار دارد. زیرا خروجی‌هایی تولید می‌کند که با استفاده از آن‌ها می‌توان، نرم‌افزار را قبل از پیاده‌سازی نهایی، ارزیابی نمود. امروزه معماران نرم‌افزار در طراحی معماری نرم‌افزار به دفعات از سبک‌های معماری استفاده می‌کنند. یک سبک معماری، ترکیبی از مولفه‌ها و ارتباطات میان آن‌ها، به همراه قواعدی درباره چگونگی ترکیب و تعامل مولفه‌ها با یکدیگر است. تاثیر سبک‌های معماری روی ویژگی‌های کیفی، معمار را قادر خواهد کرد، تصمیمات طراحی خود را با دقت و سهولت بیشتر انجام دهد. با توجه به تعدد ویژگی‌های کیفی، در این مقاله، یکی از مهمترین ویژگی‌های کیفی، یعنی قابلیت اطمینان برای ارزیابی انتخاب و سپس تاثیر سبک‌های مختلف بر روی نیاز غیروظیفه‌مندی قابلیت اطمینان ارزیابی و بررسی شد. از طرفی دیگر امروزه اکثر معماری‌هایی که برای سیستم‌های پیچیده و بزرگ طراحی می‌شوند ترکیبی از چندین سبک مختلف هستند و به صورت چندریختی ارائه می‌شوند. بنابراین سبک‌های چندریختی در میان سبک‌های معماری از جایگاه ویژه‌ای برخوردارند. لذا در این مقاله با ارائه الگوریتمی، روشی برای ارزیابی ویژگی کیفی قابلیت اطمینان در سبک‌های معماری چندریختی ارائه شده است.

کلمات کلیدی: معماری نرم‌افزار، قابلیت اطمینان، سبک‌های معماری، سبک‌های چندریختی

۱. مقدمه

با توجه به نقش مهمی که امروزه نرم‌افزار در سیستم‌ها دارد، کیفیت مولفه‌های نرم‌افزاری از اهمیت ویژه برخوردار است. یکی از جنبه‌های مهم کیفیت، قابلیت اطمینان است. ارزیابی قابلیت اطمینان، فعالیتی کلیدی و حساس در بسیاری از سیستم‌های نرم‌افزاری است که موجب کشف زود هنگام خطاها در این سیستم‌ها می‌گردد. دو دسته اصلی از مدل قابلیت اطمینان نرم‌افزار وجود دارد: مدل‌های پارامتری و مدل‌های غیرپارامتری [۲]. یکی از جنبه‌های مهم و مورد توجه در ایجاد معماری نرم‌افزار، مدل‌سازی و انتخاب سبک معماری است. یک سبک معماری یک تخصیص از انواع مولفه‌ها و ارتباطات میان آن‌ها با یکدیگر به همراه یک مجموعه قواعد و محدودیت‌ها در مورد نحوه استفاده از آن‌ها است. سبک‌های معماری نرم‌افزار به دو دسته تک‌ریختی و چندریختی تقسیم می‌شوند [۳]. برخی از انواع سبک‌های تک‌ریختی شامل: سبک ترتیبی، سبک مخزن، سبک موازی، سبک تخته سیاه، سبک لایه‌ای، سبک تحمل‌پذیر خطا، سبک سرویس‌دهنده / سرویس‌گیرنده، سبک مؤلفه‌های مستقل، سبک فراخوانی و بازگشت و غیره است [۴،۵]. مدل مبتنی بر معماری سنتی که شامل نرم‌افزارهای همگن یا تک‌ریختی هستند در اغلب موارد مناسب سیستم پیچیده نیستند لذا معمولاً در سیستم‌های نرم‌افزاری جدید از سبک‌های ناهمگن یا چندریختی استفاده می‌شود. در طراحی معماری چندریختی به جای یک سبک می‌توان، ترکیبی از چندین مولفه و سبک، برای همکاری استفاده کرد که به آن معماری مبتنی بر

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

قابلیت اطمینان ناهمگن یا چندریختی (HABR) گفته می شود. برای ارزیابی قابلیت اطمینان یک سیستم نرم افزاری ناهمگن، طراحان معماری می توانند با ترکیب چندین سبک مناسب باهم یک روش یا مدل جدیدی بسازند. اغلب سیستم ها به صورت ترکیبی از چند سبک مختلف ارائه می شوند. این چندریختی ها در انواع زیر ارائه می شوند:

چندریختی مکانی: یعنی با داشتن شکل ساختار اجرایی، الگوهایی از سبک های مختلف در مکان های مختلف به چشم می خورد. چندریختی سلسله مراتبی: یعنی یک مؤلفه از یک سبک با تقسیم شدن، طبق قواعد سبک های مختلفی ساخته شود.

چندریختی هم زمانی: یعنی هر یک از انواع مختلف سبک ها را بتوان با تعریفی از سیستم منطبق ساخت. در این نوع چندریختی، سبک های معماری نرم افزار را نمی توان به دسته های کاملاً مجزا و بدون هم پوشانی تقسیم بندی کرد [۴-۶].

یکی از مدل های پارامتری قابلیت اطمینان نرم افزار، مدل مارکوف است. نظریه پردازان مارکوف نام خود را از ریاضیدان روسی A. A. Markov (۱۸۵۲-۱۹۲۲)، که پیشگام تحقیقات سیستماتیک از توصیف فرایندهای تصادفی ریاضی است، نام بردند [۹]. مدل مارکوف یک فرآیند تصادفی زمان گسسته است که دارای ویژگی مارکوف است. فرآیند تصادفی زمان گسسته به سیستمی اشاره می کند که در هر مرحله، در حالت خاصی است و انتقال حالات بین مراحل به طور تصادفی تغییر می کند. اغلب مرحله ها را به وسیله بازه های گسسته زمانی می شناسند، اما ممکن است به مسافت فیزیکی یا معیارهای گسسته دیگری هم اشاره کند. ویژگی مارکوف بیانگر آن است که توزیع احتمال شرطی برای سیستم در مرحله بعد و در حقیقت در تمام مراحل آینده، تنها به حالت فعلی سیستم بستگی دارد و به حالات سیستم در مراحل قبلی بستگی ندارد. مدل مارکوف اغلب با یک گراف جهت دار نشان داده می شود که گره های گراف نشان دهنده حالت ها و لبه ها نشان دهنده احتمال انتقال از یک حالت به حالت های دیگر است. چون وضعیت سیستم به طور تصادفی تغییر می کند، عموماً پیش بینی حالت سیستم در آینده امکان پذیر نیست. با وجود این، ویژگی های آماری آینده سیستم را می توان پیش بینی کرد و در بسیاری کاربردها، ویژگی های آماری هستند که دارای اهمیت هستند. تغییر حالت در سیستم را انتقال و احتمالات مربوط به تغییر حالت های مختلف را احتمالات انتقال می نامند. مجموعه تمام حالت ها و احتمالات انتقال بین حالت ها در کنار هم، مدل مارکوف سیستم را ایجاد می کنند. فرض می شود که تمام حالت ها و انتقال بین آن ها در تعریف مدل مارکوف سیستم مشخص هستند، بنابراین همیشه مرحله ای بعدی هست و فرآیند مارکوف تا ابد پیش می رود [۱۳].

برای بررسی کردن قابلیت اطمینان معماری نرم افزار یک سیستم به روش مارکوف، مراحل به شرح زیر است:

برای شناسایی سبک های تکریختی معماری نرم افزار، ابتدا باید نمای معماری هر سبک را بصورت جداگانه به نمودار حالت تبدیل کرد. سپس مدل های حالت به یک مدل حالت سیستم، بر اساس معماری کلی سیستم ترکیب و ادغام خواهد شد و سبک های چندریختی را ارزیابی می کنند.

این مقاله در ادامه از بخش های زیر تشکیل میشود:

بخش دوم مروری بر کارهای قبلی، بخش سوم روش ارزیابی سبک تکریختی و سبک چندریختی، بخش چهارم نتایج شبیه سازی و بخش آخر نتیجه گیری آورده شده است.

۲. مروری بر کارهای قبلی

معماری در واقع مجموعه ای از طرح ها و تولیدات متفاوت است که سیستم را از جنبه های مختلف نشان می دهند. معماری نرم افزار از کلیدی ترین بخش های تولید نرم افزار است که البته در سال های اخیر با ایجاد مدل های کلاسیک تولید نرم افزار به نرم افزارهای عظیم تر محدود شده است. معماری نرم افزار ساختار و سازمان یک سیستم نرم افزاری است که به منظور پشتیبانی از عملیات مشخص، بر روی سازماندهی اجزاء متمرکز است. در واقع اجزاء را در حوزه های مرتبط به هم گروه بندی می کند. مهم ترین هدف در ارزیابی معماری نرم افزار، ارزیابی ویژگی های کیفی معماری است که یکی از مهم ترین آن ها قابلیت اطمینان نرم افزار است. از آن جا که معماری نرم افزار از قطعات کوچکتری به نام مولفه ها تشکیل شده اند، بنابراین قابلیت اطمینان هر یک از این مولفه ها بر روی قابلیت اطمینان معماری نرم افزار و طبیعتاً سیستم نرم افزاری که بر اساس آن توسعه می یابد بسیار حائز

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

اهمیت است. ارزیابی قابلیت اطمینان، فعالیتی کلیدی و حساس در بسیاری از سیستم‌های نرم‌افزاری است که موجب کشف زود هنگام خطاها در این سیستم‌ها می‌گردد [۱۰، ۱۱]. یکی از جنبه‌های مهم و مورد توجه در ایجاد معماری نرم‌افزار، مدل‌سازی و انتخاب سبک معماری است. یک سبک معماری یک تخصیص از انواع مولفه‌ها و ارتباطات میان آن‌ها با یکدیگر به همراه یک مجموعه قواعد و محدودیت‌ها در مورد نحوه استفاده از آن‌ها است [۱۲]. لغت سبک معماری برای اولین بار توسط Wolf Perry در سال ۱۹۹۲ معرفی شد [۱۳]. برخی از پرکاربردترین انواع سبک‌ها در معماری نرم‌افزار عبارتند از: سبک ترتیبی، مخزن، موزی، تخته سیاه، لایه‌ای، تحمل‌پذیر خطا، فراخوانی و بازگشت، مؤلفه‌های مستقل و غیره هستند.

در اغلب موارد سبک‌های تک‌ریختی مناسب سیستم پیچیده نیستند. معمولاً در سیستم‌های نرم‌افزاری جدید از سبک‌های چندریختی استفاده می‌شود که ترکیبی از چندین مولفه و سبک مختلف است. برای ارزیابی قابلیت اطمینان یک سیستم نرم‌افزاری ناهمگن، طراحان معماری می‌توانند با ترکیب چندین سبک مناسب باهم یک روش یا مدل جدیدی بسازند.

بهبود قابلیت اطمینان نرم‌افزار به طور موثر مستلزم آن است که طراحی نرم‌افزار و نقص طراحی در سریع‌ترین زمان، در چرخه زندگی توسعه نرم‌افزار شناسایی شود که باعث کاهش هزینه و تلاش می‌شود.

در طول سه دهه گذشته، تعداد زیادی از مطالعات برای اندازه‌گیری قابلیت اطمینان نرم‌افزار و بسیاری از مدل‌های تحلیلی، توسعه یافته‌اند و همچنین در مطالعات قبلی سیستم‌های نرم‌افزاری متشکل از سبک‌های مختلف معماری، عملکرد مختلف داشتند و برای تخمین و ارزیابی نرم‌افزار براساس معماری ناهمگن روش‌هایی را بررسی کردند.

در [۷] قابلیت اطمینان نرم‌افزار در مرحله طراحی، با تجزیه و تحلیل ویژگی‌های سبک‌های معماری، روابط متقابل میان مولفه‌ها با بکارگیری مدل مارکوف اندازه‌گیری شده است. در مدل ارائه شده آن‌ها از چهار سبک معماری، از جمله ترتیبی، موزی، تحمل خطا، و سبک فراخوانی و بازگشت استفاده شده است. ابتدا هر کدام از سبک‌ها را به صورت جداگانه ارزیابی شده است و سپس با ترکیبی از سبک‌ها، سبک چندریختی ایجاد شده و سپس نتیجه این مدل را در یک سیستم مالی مبتنی بر مولفه‌های صنعتی مورد ارزیابی قرار گرفته است. آنها بر این باورند که این روش مدلسازی مبتنی بر معماری، می‌تواند بسیار موثر برای بهبود کیفیت نرم‌افزار باشد.

در [۶]، مدل مبتنی بر معماری تک‌ریختی، مناسب سیستم‌های پیچیده نیستند و یک مدل مبتنی بر معماری ناهمگن ارائه و آن را بررسی کرده و در این مدل از سبک‌های ترتیبی، موزی و تحمل‌پذیر خطا استفاده شده است. در این مطالعه، نتیجه‌ی قابلیت اطمینان روش بررسی شده را در سیستم (APCMS) که بخش اصلی یک سیستم حمل بار فضا نوردی است و مسئول کنترل زیرسیستم‌ها و نظارت بر دستگاه‌های رابط است و اجزای اصلی آن مدیروظیفه، گیرنده کنترل از راه دور، مدیریت اطلاعات و سازمان داده‌ها است، با استفاده از روش مدل مبتنی بر معماری ناهمگن محاسبه کرده‌اند و به نتیجه رسیده‌اند که این مدل مناسب برای این سیستم است. زیرا نتایج به دست آمده نزدیک به قابلیت اطمینان واقعی است.

در [۱۴] درباره انواع سبک‌های معماری نرم‌افزار به صورت مجزا تحلیل و بررسی شده است از جمله سبک‌های مخزن، تخته سیاه، لایه‌ای، لوله و فیلتر، سرویس‌دهنده/سرویس‌گیرنده، کارگزار و شی‌گرا است. روش‌های مختلفی برای ارزیابی قابلیت اطمینان نرم‌افزار وجود دارد ولی در اینجا از روش دیاگرام بلوکی قابلیت اطمینان (RBD) استفاده شده و به صورت جدا هر کدام از سبک‌ها را با این روش، با فرمول‌های مختلف مورد بررسی و نتایج را در جدول‌ها و نمودارها قرار داده است.

در [۱۵] درباره (SASs) سبک معماری نرم‌افزار بر اساس میزان پشتیبانی و نگهداری که یکی از ویژگی‌های کیفیت است را بررسی کرده است. در این مطالعه، یک مدل برای تحلیل تأثیر SASs بر قابلیت نگهداری نرم‌افزار بر اساس ارزیابی مبتنی بر اندازه‌گیری SAS ارائه شده است. در این مدل، ابتدا فرمول‌ها برای محاسبه ارزش‌های اتصال، پیچیدگی و انسجام هر SAS ارائه شد. سپس مقادیر اتصال، پیچیدگی و انسجام SASها با استفاده از فرمول‌های ارائه شده کمی محاسبه شد. سپس، نسبت نسبی هر SAS در رابطه با مقادیر اتصال، پیچیدگی و انسجام SASها تعیین می‌شود. پس از آن اولویت معیارها تعیین شد و بعد رتبه نهایی تعمیر و نگهداری SASها با استفاده از روش (AHP) روند سلسله مراتب تحلیلی تعیین شد. تجزیه و تحلیل

انجام شده نشان داد که روش پیشنهادی با توجه به ارزش انواع اتصالات، مقادیر مختلف پارامتر f ، مقدار نوع انسجام و ترجیح متریک کوپلینگ به معیارهای دیگر، ثابت دارد. با ارزیابی این مقاله، نتایج نشان می دهد که اندازه گیری دقیق تر، قابل اطمینان تر و قابل تجزیه و تحلیل تر است. روش پیشنهادی فرمول هایی برای تعیین مقادیر (۱) اتصال، (۲) پیچیدگی و (۳) انسجام هر SAS می دهد، در حالی که این روش در روش های قبلی انجام نشده است.

۳. روش ارزیابی سبک تکریختی و سبک چندریختی:

در این مقاله قرار است از ۳ سبک تکریختی زیر استفاده شود:

1- سبک ترتیبی ۲- سبک موازی ۳- سبک تحمل خطا

قابلیت اطمینان معماری نرم افزار با روش مارکوف در همه سبک ها به صورت زیر محاسبه می شود.

پردازش های مارکوف به صورت گرافیکی توسط نمودارهای انتقال حالت نمایش داده می شوند. یک نمودار انتقال حالت یک گراف هدایت شده است $G=(V,E)$ که V مجموعه ای از رأس هایی است که حالت های سیستم را نشان می دهند و E مجموعه ای از لبه هایی است که نشان دهنده انتقال سیستم است. برای مدل های اطمینان، یک حالت به عنوان ترکیبی خاص از اجزای عامل و شکست خورده تعریف شده است.

برای محاسبه قابلیت اطمینان یک سیستم نمایش داده شده با یک بلوک قابلیت اطمینان، ابتدا باید سیستم را به قسمت های سریال و موازی آن تقسیم کرد. بعد، قابلیت اطمینان اجزاء محاسبه می شود.

$$R(t) = \begin{cases} \prod_{i=1}^n R_i(t), & \text{for a series structure,} \\ 1 - \prod_{i=1}^n (1 - R_i(t)), & \text{for a parallel structure.} \end{cases} \quad (1)$$

هدف از تجزیه و تحلیل فرایندهای مارکوف، محاسبه $P_i(t)$ است. $P_i(t)$ احتمال این است که سیستم در حالت i در زمان t باشد. هنگامی که $P_i(t)$ شناخته شد، قابلیت اطمینان را می توان به عنوان یک مجموع در تمام حالت های عامل محاسبه کرد. حالت ۱ یا حالت شروع را به عنوان حالتی که تمام اجزای آن ها عملی است، تعیین می کنیم. فرض کنید که در حالت ۱ سیستم، $t=0$ است.

$$P_1(0)=1$$

برای تعیین $P_i(t)$ مجموعه ای از معادلات دیفرانسیل را برای هر حالت سیستم به دست می آوریم. این معادلات، معادلات انتقال حالت نامیده می شوند که در آن انتقال از یک حالت به یک حالت دیگر تعیین می شود. معادلات انتقال حالت معمولاً در فرم ماتریسی ارائه می شود. ماتریس M که ورودی m_{ij} نرخ انتقال بین حالت های i و j است. ما از اولین شاخص i برای ستون های ماتریس و شاخص دوم j برای ردیف ها استفاده می کنیم، M دارای ساختار زیر است.

$$M = \begin{bmatrix} m_{11} & m_{21} & \dots & m_{k1} \\ m_{12} & m_{22} & \dots & m_{k2} \\ \vdots & \vdots & \vdots & \vdots \\ m_{1k} & m_{2k} & \dots & m_{kk} \end{bmatrix}$$

که در آن k تعداد حالت ها در نمودار انتقال حالت سیستم را نشان می دهد. اگر سیستم شامل n اجزاء باشد، سپس $k \leq 2^n$. جمع ورودی ها در هر ستون ماتریس انتقال باید به صفر برسند. اگر علامت هر درایه مثبت باشد به این معنا است که از حالت i شروع شده است و اگر منفی باشد یعنی به حالت i ختم می شود. با استفاده از ماتریس های انتقال حالت، معادلات انتقال حالت به صورت زیر محاسبه می شوند. اگر $P(t)$ یک بردار باشد عنصر احتمال سیستم $P_i(t)$ در حالت i در زمان t است. سپس ماتریس نمایش یک سیستم معادلات انتقال حالت به روش زیر به دست می آید.

$$\frac{d}{dt} P(t) = M.P(t) \quad (2)$$

هنگامی که سیستم معادلات حل شد و $P_i(t)$ شناخته شد، قابلیت اطمینان سیستم را می توان به عنوان یک مجموع بر روی تمام حالت های عامل محاسبه کرد [۱۲]. از آنجا که در هر زمانی سیستم می تواند تنها در حالت $P_i(0) = 0$ و $i \neq 1$ باشد

$$\sum_{i \in OUF} p_i(t) = 1 \quad (3)$$

جایی که مجموع بر روی تمام حالت های عامل O انجام می شود. قابلیت اطمینان را می توان به صورت زیر محاسبه کرد.

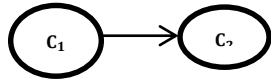
سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

$$R(t) = 1 - \sum_{i \in F} p_i(t) \quad (4)$$

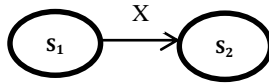
در ادامه به منظور درک بهتر روابط قبلی، چند مثال بررسی می شود.

سبک ترتیبی:

در این سبک، مؤلفه ها به صورت ترتیبی اجرا می شوند، به طوری که در هر لحظه ای از زمان، تنها یک مؤلفه در حال اجرا است و پس از تمام شدن کار آن مؤلفه به مؤلفه بعدی منتقل می شود. در این سبک گام های پردازش یا همان مؤلفه ها، برنامه های مستقلی هستند و فرض بر این است که هر گام قبل از شروع گام بعد، اجرائیش به تکامل می رسد. هر دسته ای از داده ها، بین گام ها انتقال داده می شوند. با فرض این که معماری از K مؤلفه تشکیل شده باشد K حالت در زنجیره مارکوف مربوط به این سبک ها وجود دارد.



الف) نمای معماری



ب) نمودار حالت

شکل (1) نمای معماری و نمودار حالت سبک ترتیبی

ماتریس احتمال انتقال آن به صورت زیر ایجاد می شود.

$$M = \begin{bmatrix} -x & 0 \\ x & 0 \end{bmatrix}$$

سپس ماتریس به دست آمده داخل فرمول زیر قرار داده می شود.

$$\frac{d}{dt} P(t) = M \cdot P(t)$$

$$\frac{d}{dt} \begin{bmatrix} p_1(t) \\ p_2(t) \end{bmatrix} = \begin{bmatrix} -x & 0 \\ x & 0 \end{bmatrix} \begin{bmatrix} p_1(t) \\ p_2(t) \end{bmatrix}$$

سپس $p_i(t)$ ها از رابطه بالا به دست می آید.

$$\frac{d}{dt} p_1(t) = -x p_1(t) \dots \dots \dots p_1(t) = e^{-x_1 t}$$

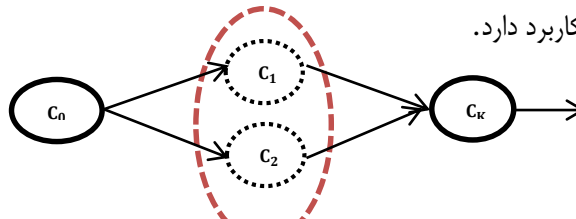
$$\frac{d}{dt} p_2(t) = x p_1(t) \dots \dots \dots p_2(t) = e^{x_1 t}$$

این سبک از نوع سریال می باشد. زیرا اگر یکی از مؤلفه ها خراب شود کل سیستم خراب می شود. پس در این سیستم، فقط حالت اول جواب می شود.

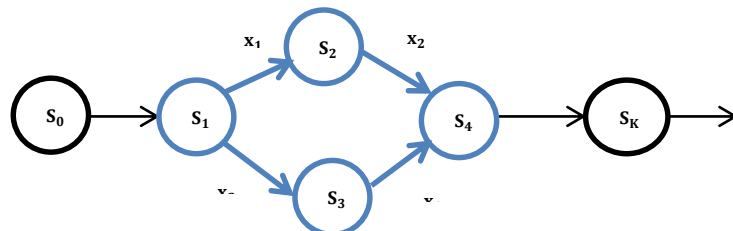
$$R(t) = e^{-x_1 t}$$

سبک موازی:

معمولا برای دستیابی به سرعت بالاتر در محاسبات، از سبک موازی استفاده می شود. این سبک دارای مؤلفه های متعدد در حال اجرا به صورت همزمان است. از این رو زمان پاسخ کاهش می یابد و این موضوع باعث بهبود قابلیت اطمینان خواهد شد. سبک موازی شبیه سبک لوله و فیلتر است. هر دو سبک در محیط های اجرای موازی کاربرد دارند. با این تفاوت که سبک موازی در محیط هایی که دارای چند پردازنده است استفاده می شود در حالی که سبک لوله و فیلتر در محیط هایی که چندین پردازش روی یک پردازنده اجرا می شوند، کاربرد دارد.

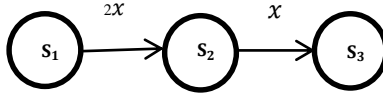


الف) نمای معماری



سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

ب) نمودار حالت



ج) ساده شده نمودار حالت

شکل ۲) نمای معماری و نمودار حالت و ساده شده نمودار حالت سبک موازی

حال ماتریس احتمال انتقال آن به صورت زیر به دست خواهد آمد.

$$M = \begin{bmatrix} -2x & 0 & 0 \\ 2x & -x & 0 \\ 0 & x & 0 \end{bmatrix}$$

سپس ماتریس به دست آمده داخل فرمول قرار داده می شود.

$$\frac{d}{dt} \begin{bmatrix} p_1(t) \\ p_2(t) \\ p_3(t) \end{bmatrix} = \begin{bmatrix} -2x & 0 & 0 \\ 2x & -x & 0 \\ 0 & x & 0 \end{bmatrix} \cdot \begin{bmatrix} p_1(t) \\ p_2(t) \\ p_3(t) \end{bmatrix}$$

سپس $P_1(t)$ ها از رابطه بالا به دست می آید.

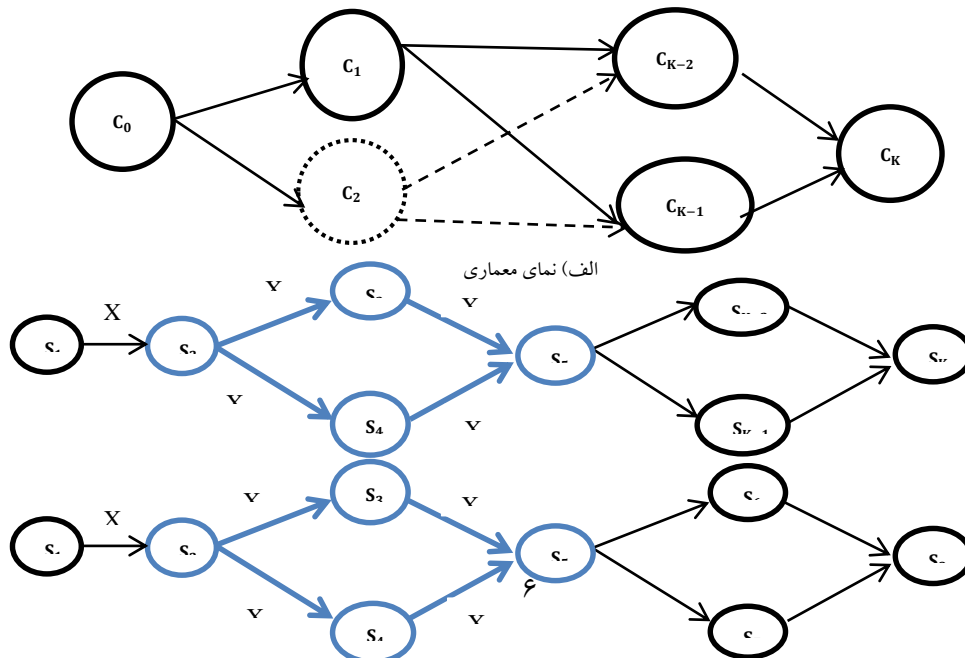
$$\begin{aligned} \frac{d}{dt} p_1(t) &= -2xp_1(t) \dots\dots\dots p_1(t) = e^{-2tx} \\ \frac{d}{dt} p_2(t) &= 2xp_1(t) - xp_2(t) \dots\dots\dots p_2(t) = 2e^{(-tx)} - 2e^{(-2tx)} \\ \frac{d}{dt} p_3(t) &= xp_2(t) \dots\dots\dots p_3(t) = e^{(-2tx)} - 2e^{(-tx)} + 1 \end{aligned}$$

این سبک از نوع سریال است. اگر یکی از مؤلفه ها خراب شود کل سیستم خراب می شود چون به صورت همزمان کار می کنند. در این سیستم موازی، مؤلفه ها پردازش های یکسان انجام می دهند. پس در این سیستم، حالت اول جواب می شود.

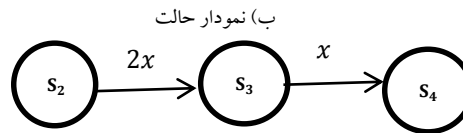
$$R(t) = e^{-2xt}$$

سبک تحمل خطا:

این سبک معماری شامل مؤلفه های اصلی و مجموعه ای از مؤلفه های پشتیبان است که می تواند با الگوریتم های متفاوت و یا در ساختارهای گوناگون با توجه به مؤلفه های اصلی خود پیاده سازی شود. مؤلفه های این سبک معماری که شامل مؤلفه های اصلی و پشتیبان هستند و به صورت موازی قرار می گیرند تا زمانی که یک مؤلفه از مؤلفه های اصلی از کار افتاد و یا دچار اشکال شد، با جایگزاری از مؤلفه های پشتیبان سیستم بتواند همچنان سرویس ارائه نماید. در نمای معماری ارائه داده شده در شکل (الف) مؤلفه خط چین به عنوان مؤلفه پشتیبان برای مؤلفه اصلی C_2 است.



سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم



(ج) ساده شده نمودار حالت

شکل ۳) نمای معماری و نمودار حالت و ساده شده نمودار حالت سبک تحمل خطا

ماتریس احتمال انتقال آن به صورت زیر است.

$$M = \begin{bmatrix} -2x & 0 & 0 \\ 2x & -x & 0 \\ 0 & x & 0 \end{bmatrix}$$

سپس ماتریس به دست آمده داخل فرمول قرار داده می شود.

$$\frac{d}{dt} \begin{bmatrix} p_1(t) \\ p_2(t) \\ p_3(t) \end{bmatrix} = \begin{bmatrix} -2x & 0 & 0 \\ 2x & -x & 0 \\ 0 & x & 0 \end{bmatrix} \cdot \begin{bmatrix} p_1(t) \\ p_2(t) \\ p_3(t) \end{bmatrix}$$

سپس $P_i(t)$ ها از رابطه بالا به دست می آید.

$$\begin{aligned} \frac{d}{dt} p_1(t) &= -2xp_1(t) \dots\dots\dots p_1(t) = e^{-2tx} \\ \frac{d}{dt} p_2(t) &= 2xp_1(t) - xp_2(t) \dots\dots\dots p_2(t) = 2e^{-tx} - 2e^{-2tx} \\ \frac{d}{dt} p_3(t) &= xp_2(t) \dots\dots\dots p_3(t) = e^{-2tx} - 2e^{-tx} + 1 \end{aligned}$$

این سبک از نوع موازی است. اگر یکی از مولفه ها خراب شود کل سیستم خراب نمی شود بلکه کارشان را ادامه می دهند چون هر مولفه ای، مولفه پشتیبان دارد و در زمان خرابی جایگزین آن مولفه می شود. پس در این سیستم، مجموع حالت اول و دوم جواب است.

$$R(t) = e^{-2tx} + 2e^{-tx} - 2e^{-2tx}$$

در بخش قبل نشان دادیم که چگونه قابلیت اطمینان را برای سیستمی بر مبنای یک سبک معماری محاسبه نمایم در این بخش می خواهیم قابلیت اطمینان را برای معماری که از سبک های چندریختی حاصل شده است محاسبه کنیم که به صورت زیر انجام می گیرد.

الف) سبک های موجود در معماری سیستم، بر اساس مشخصات طراحی سیستم تعیین می شود.

ب) نمای معماری هر سبک موجود را به نمودار حالت تبدیل کرده و قابلیت اطمینان، برای هر نمودار حالت محاسبه می شود.

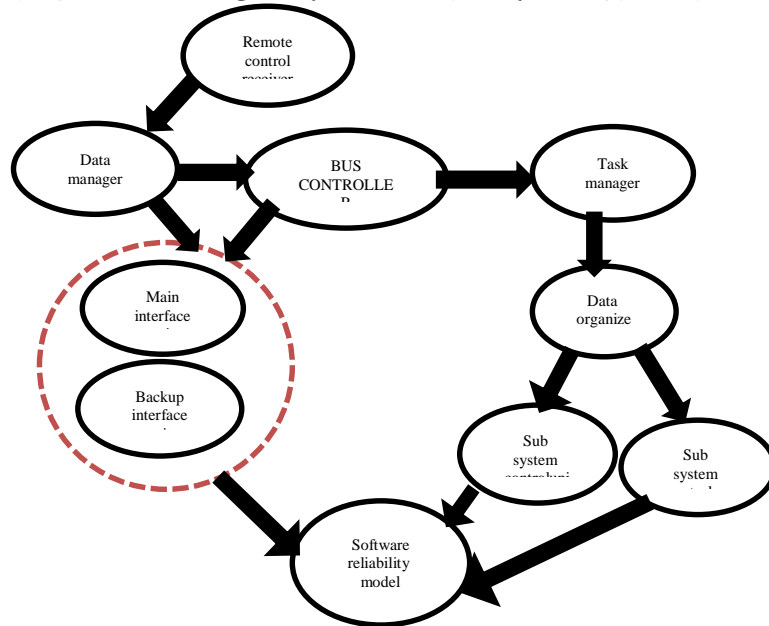
ج) جدول حالت سراسری سیستم، با توجه به نمودار حالت موجود بر اساس نمای معماری کامل سیستم ترسیم می شود.

د) ماتریس انتقال حالت را برای نمودار حالت سراسری سیستم ایجاد کرده و قابلیت اطمینان را برای کل سیستم با توجه به روابط ذکر شده و به دست آوردن $p_i(t)$ ها، محاسبه می شود.

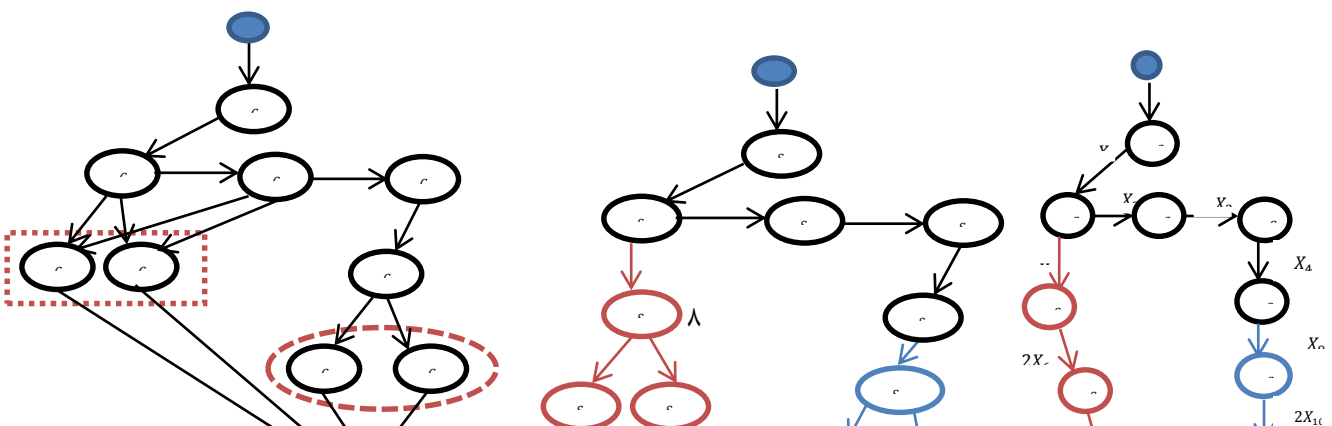
برای مثال، قابلیت اطمینان سیستم APCMS محاسبه می شود.

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

این سیستم، بخش اصلی سیستم حمل بار فضاوردی است که مسئول کنترل زیرسیستمها و نظارت بر دستگاههای رابط است. اجزای اصلی آن مدیروظیفه، گیرنده کنترل از راه دور، مدیریت اطلاعات و سازمان دادهها است. شکل زیر این سیستم را نشان می دهد.



مرحله اول: این سیستم از ۱۰ مولفه تشکیل شده و این مولفهها از سه سبک تکریختی تحمل پذیرخطا، موازی و ترتیبی ترکیب شدند. که c5, c6 سبک تحمل پذیرخطا و c8, c9 سبک موازی و بقیه مولفهها از سبک ترتیبی هستند. مرحله دوم: نمای معماری سیستم را به نمودار حالت تبدیل کرده به این صورت که c5, c6 به c8, c9 و s5 به s7 تبدیل می شود. مرحله سوم: جدول حالت سیستم را رسم کرده. چون ۸ مولفه در نمودار حالت است پس $2^8 = 256$ حالت در جدول خواهد بود. مرحله چهارم: ماتریس انتقال حالت را برای نمودار حالت سراسری سیستم ایجاد کرده. مرحله پنجم: طبق فرمول، $p_i(t)$ هارا به دست آورده و قابلیت اطمینان، حساب می شود.



سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

(ج) ساده شده نمودار حالت (ب) نمودار حالت الف) نمای معماری
شکل ۴) نمای معماری و نمودار حالت و ساده شده نمودار حالت سبک چندریختی

مقادیر قابلیت اطمینان اجزای داده شده عبارتند از:

$$\begin{matrix} X_1 = 0/40 & X_2 = 0/10 & X_3 = 0/20 & X_4 = 0/80 & X_5 = 0/90 & X_6 = 0/45 \\ X_7 = 0/35 & X_8 = 0/30 & X_9 = 0/65 & X_{10} = 0/60 & X_{11} = 0/75 & X_{12} = 1 \end{matrix}$$

ماتریس انتقال برای نمودار انتقال حالت که در شکل ۳ نشان داده شده است:

$$M = \begin{bmatrix} -x_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_1 & -x_2 - x_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & x_2 & -x_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & x_3 & -x_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_4 & -x_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & x_5 & 0 & 0 & 0 & -2x_6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2x_6 & -x_7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_7 & -x_8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_9 & 0 & 0 & 0 & -2x_{10} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2x_{10} & -x_{11} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_{11} & -x_{12} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_8 & 0 & 0 & x_{12} & 0 \end{bmatrix}$$

$$\frac{d}{dt} P(t) = M.P(t)$$

$$\frac{d}{dt} \begin{bmatrix} p_1(t) \\ p_2(t) \\ p_3(t) \\ p_4(t) \\ p_5(t) \\ p_6(t) \\ p_7(t) \\ p_8(t) \\ p_9(t) \\ p_{10}(t) \\ p_{11}(t) \\ p_{12}(t) \end{bmatrix} = \begin{bmatrix} -x_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_1 & -x_2 - x_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & x_2 & -x_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & x_3 & -x_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_4 & -x_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & x_5 & 0 & 0 & 0 & -2x_6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2x_6 & -x_7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_7 & -x_8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_9 & 0 & 0 & 0 & -2x_{10} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2x_{10} & -x_{11} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_{11} & -x_{12} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_8 & 0 & 0 & x_{12} & 0 \end{bmatrix} \begin{bmatrix} p_1(t) \\ p_2(t) \\ p_3(t) \\ p_4(t) \\ p_5(t) \\ p_6(t) \\ p_7(t) \\ p_8(t) \\ p_9(t) \\ p_{10}(t) \\ p_{11}(t) \\ p_{12}(t) \end{bmatrix}$$

رابطه‌ها	جواب‌ها
$\frac{d}{dt} p_1(t) = -x_1 p_1(t)$	$p_1(t) = e^{-(0.4*t)}$
$\frac{d}{dt} p_2(t) = x_1 p_1(t) + (-x_2 - x_5) p_2(t)$	$p_2(t) = \frac{2}{7} * e^{-t} - \frac{2 * e^{-t} * e^{-\frac{7t}{5}}}{7}$

$\frac{d}{dt} p_3(t) = x_2 p_2(t) - x_3 p_3(t)$ $\frac{d}{dt} p_4(t) = x_3 p_3(t) - x_4 p_4(t)$ $\frac{d}{dt} p_5(t) = x_4 p_4(t) - x_9 p_5(t)$ $\frac{d}{dt} p_6(t) = x_5 p_2(t) - 2x_6 p_6(t)$ $\frac{d}{dt} p_7(t) = 2x_6 p_6(t) - x_7 p_7(t)$ $\frac{d}{dt} p_8(t) = x_7 p_7(t) + x_8 p_8(t)$ $\frac{d}{dt} p_9(t) = x_9 p_5(t) - 2x_{10} p_9(t)$ $\frac{d}{dt} p_{10}(t) = 2x_{10} p_9(t) - x_{11} p_{10}(t)$ $\frac{d}{dt} p_{11}(t) = x_{11} p_{10}(t) - x_{12} p_{11}(t)$ $\frac{d}{dt} p_{12}(t) = x_{12} p_{11}(t)$	$p_3(t) = \frac{e^{-\frac{t}{5}}}{12} + e^{-\frac{t}{5}} * (\frac{e^{-\frac{3t}{5}}}{21} + \frac{e^{-\frac{4t}{5}}}{28})$ $p_4(t) = \frac{e^{-\frac{4t}{5}}}{18} - \frac{e^{-\frac{t}{5}} * e^{-\frac{4t}{5}} * (7 * e^{-\frac{4t}{5}} - 2 * e^{-\frac{7t}{5}} + 9)}{252}$ $p_5(t) = \frac{1024}{3969} * e^{-\frac{13t}{20}} + e^{-\frac{13t}{20}} * (\frac{8 * e^{-\frac{13t}{20}}}{27} - \frac{4 * e^{-\frac{7t}{20}}}{49} + \frac{4 * e^{-\frac{9t}{20}}}{81} - \frac{8 * e^{-\frac{21t}{20}}}{1323})$ $p_6(t) = -\frac{36 * e^{-\frac{9t}{10}}}{13} + \frac{18 * e^{-\frac{t}{10}} * e^{-\frac{9t}{10}} * (e^{-\frac{7t}{5}} + 13)}{91}$ $p_7(t) = -\frac{864 * e^{-\frac{7t}{20}}}{715} + \frac{108 * e^{-\frac{7t}{20}} * e^{-\frac{13t}{20}} * (210 * e^{-\frac{t}{10}} + 11 * e^{-\frac{7t}{5}} - 165)}{5005}$ $p_8(t) = e^{-\frac{3t}{10}} * (\frac{54 * e^{\frac{t}{10}}}{65} - \frac{189 * e^{-\frac{6t}{5}}}{143} + \frac{162 * e^{-\frac{13t}{10}}}{169} + \frac{6048 * e^{-\frac{13t}{20}}}{9295}) - \frac{189 * e^{-\frac{3t}{10}}}{169}$ $p_9(t) = -\frac{13 * e^{-\frac{6t}{5}}}{220} + e^{-\frac{6t}{5}} * (\frac{13 * e^{\frac{t}{5}}}{49} - \frac{13 * e^{-\frac{2t}{5}}}{27} + \frac{13 * e^{-\frac{8t}{5}}}{5292} + \frac{13312 * e^{-\frac{11t}{20}}}{43659} - \frac{13 * e^t}{405})$ $p_{10}(t) = -\frac{53248 * e^{-\frac{3t}{4}}}{3795} + e^{-\frac{3t}{4}} * (\frac{53248 * e^{\frac{t}{10}}}{14553} - \frac{312 * e^{-\frac{t}{4}}}{245} + \frac{104 * e^{-\frac{t}{20}}}{9} + \frac{26 * e^{-\frac{9t}{20}}}{165} - \frac{104 * e^{-\frac{11t}{20}}}{1485} + \frac{26 * e^{-\frac{23t}{20}}}{10143})$ $p_{11}(t) = -\frac{256 * e^{-\frac{t}{10}}}{847} + e^{-\frac{t}{10}} * (\frac{13 * e^{\frac{t}{10}}}{3381} + \frac{52 * e^{-\frac{t}{10}}}{99} - \frac{260 * e^{-\frac{7t}{20}}}{21} + \frac{52 * e^{-\frac{9t}{10}}}{49} - \frac{13 * e^{-\frac{11t}{10}}}{121} + \frac{266240 * e^{-\frac{11t}{20}}}{53361} + \frac{4096 * e^{-\frac{13t}{20}}}{253})$ $p_{12}(t) = \frac{13 * e^{\frac{2t}{5}}}{13524} - \frac{26 * e^{-\frac{t}{5}}}{99} - \frac{8192 * e^{-\frac{3t}{4}}}{3795} - \frac{26 * e^{-t}}{245} + \frac{65 * e^{-\frac{4t}{5}}}{42} + \frac{13 * e^{-\frac{6t}{5}}}{1452} + \frac{256 * e^{-\frac{t}{10}}}{847} - \frac{40960 * e^{-\frac{13t}{20}}}{53361} - \frac{1}{10}$
---	--

در اینجا به دلیل اینکه سیستم موازی است، جواب نهایی، مجموع $p_1(T)$ تا $p_{11}(T)$ است.

۱- نتایج شبیه سازی:

روش پیشنهادی در قسمت قبل کامل توضیح داده شد. حال قرار است این روش با روش رویکرد مبتنی بر مارکوف مقایسه شود.

۴-۱- روش رویکرد مبتنی بر مارکوف:

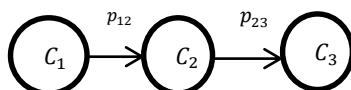
در روش رویکرد مبتنی بر مارکوف، قابلیت اطمینان معماری نرم افزار با روش مارکوف در همه سبکها (هم تک ریختی هم چند ریختی) به صورت زیر محاسبه می شود.

ابتدا نمای معماری به نمودار حالت تبدیل شده سپس از روی نمودار حالت، ماتریس انتقال آن سبک، طبق فرمول هر سبک و با توجه به مقادیر ورودی قابلیت اطمینان و مقادیر ورودی حالت انتقال بین مولفه ها به وجود می آید. سپس قابلیت اطمینان سراسری کل سیستم طبق فرمول محاسبه می شود [۷، ۱].

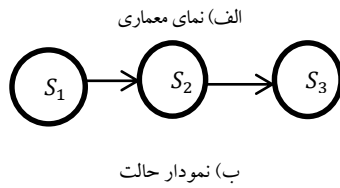
محاسبه سبک تک ریختی:

سبک ترتیبی:

در این سبک، مولفه ها به صورت ترتیبی اجرا می شوند، به طوری که در هر لحظه ای از زمان، تنها یک مولفه در حال اجرا است و پس از تمام شدن کار آن مولفه به مولفه بعدی منتقل می شود. در این سبک گام های پردازش یا همان مولفه ها، برنامه های مستقلی هستند و فرض بر این است که هر گام قبل از شروع گام بعد، اجرائیش به تکامل می رسد. هر دسته ای از داده ها، بین گامها انتقال داده می شوند.



سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم



شکل ۵) نمودار حالت و نمای معماری سبک ترتیبی

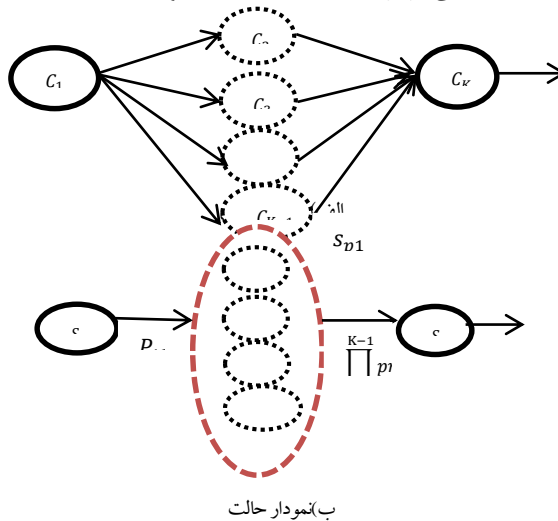
در این سبک ماتریس انتقال M به صورت زیر تشکیل می شود:

$$M(i, j) = R_i P_{ij} \quad \text{از } S_i \text{ به } S_j \text{ می رود} \quad (5)$$

$$M(i, j) = 0 \quad \text{از } S_i \text{ به } S_j \text{ نمی رود}$$

سبک موازی:

در این سبک، چندین مؤلفه می توانند همزمان اجرا شوند. همانطور که در شکل دیده می شود چنانچه K مؤلفه در این معماری موجود باشند و $L=K-2$ مؤلفه آن همزمان اجرا شوند در این صورت این تعداد مؤلفه در نمودار حالت، در یک حالت قرار می گیرند و لذا تعداد کل حالت های موجود $K-L+1$ خواهد بود. بخاطر خصوصیت های سبک موازی احتمالات انتقال از مؤلفه C_1 به C_2, C_3, \dots, C_{K-1} همگی برابر P_{12} هستند که در نمودار حالت از حالت S_1 به SP_1 رفته است.



شکل ۶) نمودار حالت و نمای معماری سبک موازی

برای محاسبه $M(\{SP_1\}, \{S_K\})$ لازم است تا کلیه احتمالات انتقالی از C_{K-1}, C_3, C_2 که در حالت SP_1 است به وضعیت S_K در نظر بگیریم و چون قابلیت اطمینان مؤلفه ها و احتمالات انتقال هر کدام از یکدیگر مستقل هستند بنابراین مقدار $M(\{SP_1\}, \{S_K\})$ به صورت زیر محاسبه می شود:

$$M(\{SP_1\}, \{S_K\}) = \prod_{n=2}^{K-1} R_n P_{nk}$$

و بنابراین ماتریس M به صورت زیر بدست می آید:

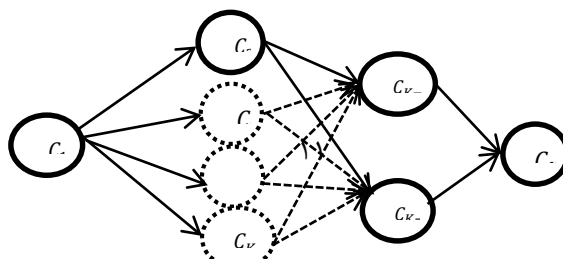
$$M(i, j) = R_i P_{ij} \quad S_i \notin S_p \quad (6)$$

$$M(i, j) = \prod R_n P_{nj}, \quad S_i \in S_p$$

$$M(i, j) = 0 \quad \text{از } S_i \text{ به } S_j \text{ نمی رود}$$

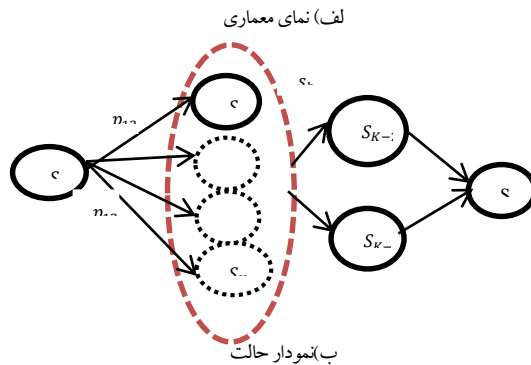
سبک تحمل خطا:

این سبک معماری شامل مؤلفه های اصلی و مجموعه ای از مؤلفه های پشتیبان است که می تواند با الگوریتم های متفاوت و یا در ساختارهای گوناگون با توجه به مؤلفه های اصلی خود پیاده سازی شود. در نمای معماری ارائه شده در شکل



سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

زیر مؤلفه‌های خط چین از C_3 تا C_{K-1} به عنوان مؤلفه‌های پشتیبان برای مؤلفه اصلی C_2 هستند که در نمودار حالت شکل زیر در حالت S_b قرار گرفته‌اند.



شکل ۷) نمودار حالت و نمای معماری سبک تحمل خطا

فرض کنید که تمام مؤلفه‌های پشتیبان احتمالات انتقال یکسان با مؤلفه اصلی دارند. چنانچه معماری شامل K مؤلفه باشد که $L=K-4$ تا از آن‌ها به صورت پشتیبان عمل می‌کنند در این صورت $K-L+1$ حالت در نمودار حالت وجود خواهد داشت. در نمودار حالت مربوط به این سبک هرگاه S_2 از کار بیفتد S_3 جایگزین آن شده و باعث بهبود قابلیت اطمینان می‌شود چنانچه S_2 و S_3 از کار بیفتند این پشتیبانی را S_4 انجام می‌دهد، به همین ترتیب تا S_{K-3} قابل محاسبه به صورت زیر است:

$$R_{b1} = R_2 + \sum_{n=3}^{K-3} ((\prod_{m=2}^{n-1} (1 - R_m)) R_n)$$

اگر تمام مؤلفه‌های پشتیبان احتمالات انتقال یکسان با مؤلفه اصلی داشته باشند مقدار $M(\{SP_1\}, \{S_{K-1}\})$ و $M(\{SP_1\}, \{S_{K-2}\})$ به ترتیب برابر $R_{b1} P_{2(K-2)}$ و $R_{b1} P_{2(K-1)}$ خواهد بود.

بنابراین ماتریس انتقال M به صورت زیر بدست می‌آید:

$$M(i, j) = R_i P_{ij} \quad S_i \notin S_b \quad (V)$$

$$M(i, j) = (R_{a1} + \sum_{q=a2}^{ar} ((\prod_{m=a1}^{q-1} (1 - R_m)) R_q)) P_{a1,j}, \quad S_i \in S_b$$

$$M(i, j) = 0 \quad \text{از } S_i \text{ به } S_j \text{ نمی‌رود}$$

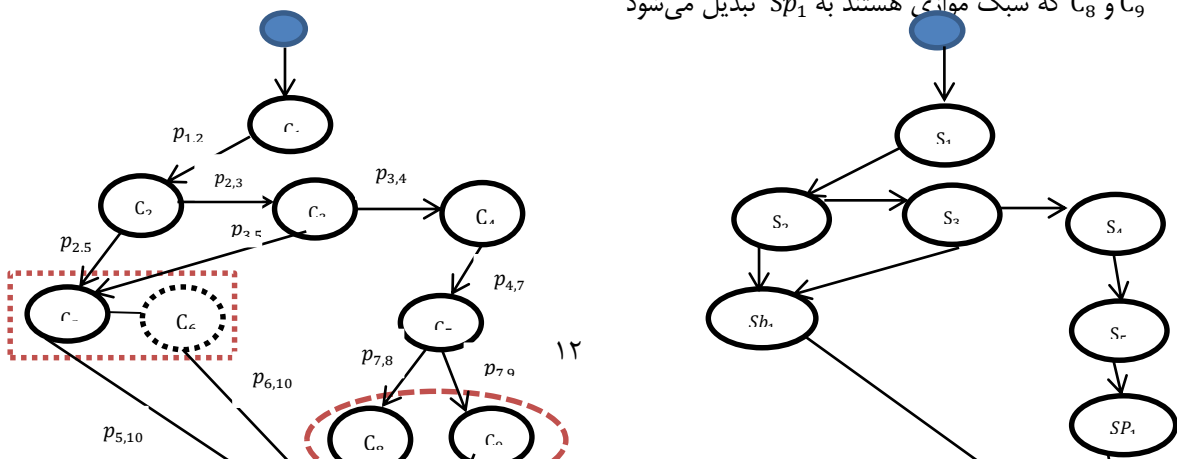
محاسبه سبک چندریختی:

اکنون که توضیحات، درباره ۳ سبک گفته شد، در ادامه برای فهم بهتر مثالی زده می‌شود:

سیستم زیر را در نظر بگیرید:

در مرحله اول نمای معماری به نمودار حالت تبدیل می‌شود یعنی C_5 و C_6 مربوط به سبک تحمل خطا هست به S_{b1} و مؤلفه‌های

C_8 و C_9 که سبک موازی هستند به SP_1 تبدیل می‌شود



سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

الف) نمای معماری

ب) نمودار حالت

شکل ۸) نمای معماری و نمودار حالت سبک چندریختی

در اینجا قابلیت اطمینان کل سیستم محاسبه می شود. قابلیت اطمینان مولفه ها با R_i و احتمال انتقال بین مولفه ها با $P_{i,j}$ نشان داده می شود.

$$\begin{array}{llllll}
 P_{1,2} = 1 & P_{2,3} = 0.75 & P_{3,4} = 0.7 & P_{2,5} = 0.15 & P_{3,5} = 0.08 & P_{4,7} = 1 \\
 P_{7,8} = 0.5 & P_{7,9} = 0.5 & P_{8,10} = 0.5 & P_{9,10} = 0.5 & P_{5,10} = 0.125 & P_{6,10} = 0.1 \\
 R_1 = 0.75 & R_2 = 0.7 & R_3 = 0.7 & R_4 = 0.5 & R_5 = 0.7 & R_6 = 0.7 \\
 R_7 = 0.75 & R_8 = 0.7 & R_9 = 0.7 & R_{10} = 0.7 & &
 \end{array}$$

در این مرحله طبق فرمول های هر سبک، ماتریس سیستم به دست می آید.

$$M = \begin{bmatrix}
 0 & 0.75 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0.05 & 0 & 0.15 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0.02 & 0.08 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0.50 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.125 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
 \end{bmatrix}$$

قابلیت اطمینان سراسری کل سیستم به صورت زیر محاسبه می شود:

$$R = T_{(1,n)} * R_n \quad (۸)$$

$$T_{(1,n)} = (-1)^{K+1} \cdot \frac{|E|}{|I-M|} \quad (۹)$$

I : یک ماتریس $k \times k$ است.

M : یک ماتریس انتقال حالت است.

$|E|$: دترمینان ماتریس حاصل از حذف آخرین ردیف و اولین ستون ماتریس $(I - M)$

$|I-M|$: دترمینان ماتریس $(I - M)$ که فرض می شود.

$$|I - M| = \begin{bmatrix}
 1 & -0.75 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & -0.05 & 0 & -0.15 & 0 & 0 & 0 \\
 0 & 0 & 1 & -0.02 & -0.08 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & -0.50 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & -0.125 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{bmatrix} = 1$$

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

$$|E| = \begin{bmatrix} -0.75 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -0.05 & 0 & -0.15 & 0 & 0 & 0 \\ 0 & 1 & -0.02 & -0.08 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -0.50 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -0.125 \end{bmatrix} = 0.0156$$

$$R = ((-1)^{8+1} \cdot \frac{0.0156}{1}) * 0/2 = -0.0031$$

حال این دو روش پیشنهاد شده و روش رویکرد مبتنی بر مارکوف با مثال زیر با هم مقایسه می شوند.

جدول های زیر ۴ حالت از مقادیر احتمال انتقال بین مولفه ها در روش رویکرد مبتنی بر مارکوف و روش پیشنهادی برای تست آورده شده است:

احتمال انتقال در روش پیشنهادی و روش رویکرد مبتنی بر مارکوف و قابلیت اطمینان مولفه ها در حالت اول:

$$\begin{array}{llllll} X_1 = 0/005 & X_2 = 0/0001 & X_3 = 0/002 & X_4 = 0/0007 & X_5 = 0/008 & X_6 = 0/001 \\ X_7 = 0/0001 & X_8 = 0/0002 & X_9 = 0/004 & X_{10} = 0/004 & X_{11} = 0/007 & X_{12} = 0/003 \\ P_{1,2} = 1 & P_{2,3} = 0/15 & P_{3,4} = 0/25 & P_{2,5} = 0/85 & P_{3,5} = 0/75 & P_{4,7} = 1 \\ P_{7,8} = 0/5 & P_{7,9} = 0/5 & P_{8,10} = 0/5 & P_{9,10} = 0/5 & P_{5,10} = 0/35 & P_{6,10} = 0/65 \\ R_1 = 0/5 & R_2 = 0/3 & R_3 = 0/9 & R_4 = 0/8 & R_5 = 0/25 & R_6 = 0/9 \\ R_7 = 0/1 & R_8 = 0/9 & R_9 = 0/65 & R_{10} = 0/4 & & \end{array}$$

احتمال انتقال در روش پیشنهادی و روش رویکرد مبتنی بر مارکوف و قابلیت اطمینان مولفه ها در حالت دوم:

$$\begin{array}{llllll} X_1 = 0/009 & X_2 = 0/0001 & X_3 = 0/004 & X_4 = 0/007 & X_5 = 0/008 & X_6 = 0/001 \\ X_7 = 0/0001 & X_8 = 0/0003 & X_9 = 0/006 & X_{10} = 0/006 & X_{11} = 0/007 & X_{12} = 0/007 \\ P_{1,2} = 1 & P_{2,3} = 0/3 & P_{3,4} = 0/6 & P_{2,5} = 0/7 & P_{3,5} = 0/4 & P_{4,7} = 1 \\ P_{7,8} = 0/5 & P_{7,9} = 0/5 & P_{8,10} = 0/5 & P_{9,10} = 0/5 & P_{5,10} = 0/35 & P_{6,10} = 0/65 \\ R_1 = 0/3 & R_2 = 0/4 & R_3 = 0/55 & R_4 = 0/1 & R_5 = 0/6 & R_6 = 0/7 \\ R_7 = 0/45 & R_8 = 1 & R_9 = 0/3 & R_{10} = 0/8 & & \end{array}$$

احتمال انتقال در روش پیشنهادی و روش رویکرد مبتنی بر مارکوف و قابلیت اطمینان مولفه ها در حالت سوم:

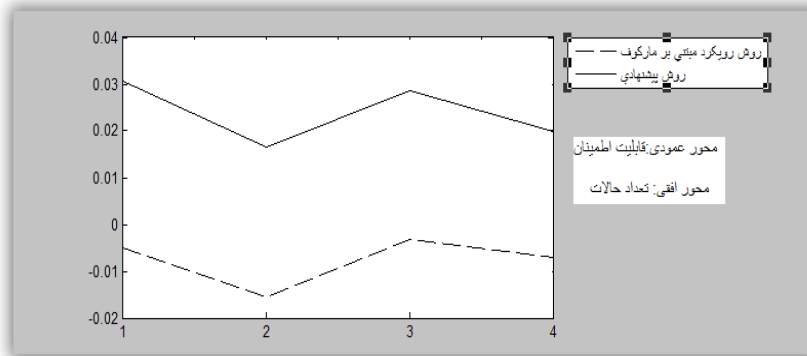
$$\begin{array}{llllll} X_1 = 0/008 & X_2 = 0/0001 & X_3 = 0/006 & X_4 = 0/0007 & X_5 = 0/008 & X_6 = 0/001 \\ X_7 = 0/0003 & X_8 = 0/0005 & X_9 = 0/006 & X_{10} = 0/004 & X_{11} = 0/006 & X_{12} = 0/007 \\ P_{1,2} = 1 & P_{2,3} = 0/25 & P_{3,4} = 0/2 & P_{2,5} = 0/75 & P_{3,5} = 0/8 & P_{4,7} = 1 \\ P_{7,8} = 0/5 & P_{7,9} = 0/5 & P_{8,10} = 0/5 & P_{9,10} = 0/5 & P_{5,10} = 0/15 & P_{6,10} = 0/85 \\ R_1 = 0/75 & R_2 = 0/2 & R_3 = 0/1 & R_4 = 0/5 & R_5 = 0/9 & R_6 = 0/1 \\ R_7 = 0/25 & R_8 = 0/6 & R_9 = 0/9 & R_{10} = 0/1 & & \end{array}$$

احتمال انتقال در روش پیشنهادی و روش رویکرد مبتنی بر مارکوف و قابلیت اطمینان مولفه ها در حالت چهارم:

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

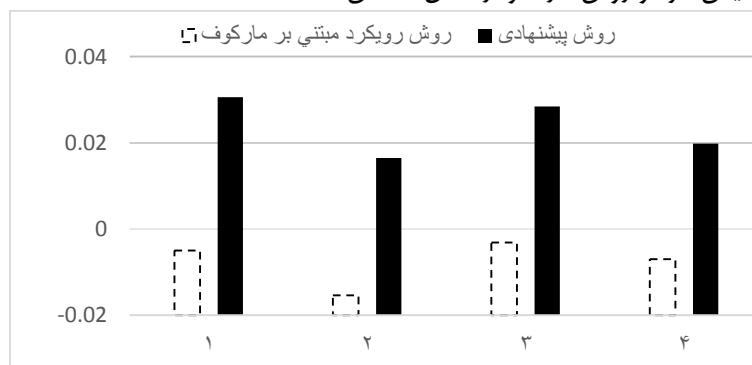
$$\begin{aligned}
 X_1 &= 0/007 & X_2 &= 0/0001 & X_3 &= 0/002 & X_4 &= 0/0008 & X_5 &= 0/009 & X_6 &= 0/001 \\
 X_7 &= 0/0002 & X_8 &= 0/0003 & X_9 &= 0/006 & X_{10} &= 0/006 & X_{11} &= 0/007 & X_{12} &= 0/007 \\
 P_{1,2} &= 1 & P_{2,3} &= 0/6 & P_{3,4} &= 0/3 & P_{2,5} &= 0/4 & P_{3,5} &= 0/7 & P_{4,7} &= 1 \\
 P_{7,8} &= 0/5 & P_{7,9} &= 0/5 & P_{8,10} &= 0/5 & P_{9,10} &= 0/5 & P_{5,10} &= 0/8 & P_{6,10} &= 0/2 \\
 R_1 &= 0/9 & R_2 &= 0/1 & R_3 &= 0/7 & R_4 &= 0/5 & R_5 &= 0/4 & R_6 &= 0/5 \\
 R_7 &= 0/45 & R_8 &= 0/1 & R_9 &= 0/9 & R_{10} &= 0/35
 \end{aligned}$$

شکل ۹ نمودار قابلیت اطمینان هر دو روش با ۴ حالت مختلف را برای مثال نشان می‌دهد. این ۴ حالت، جداول ۳ تا ۱۴ هستند و همان‌طور که ملاحظه می‌شود با مقادیر متفاوت مؤلفه‌ها و با ورودی‌های مختلفی که برای این ۴ حالت در نظر گرفته شده، قابلیت اطمینان‌ها (محور عمودی) در هر دو روش نزدیک به هم هستند.



شکل ۹) نمودار قابلیت اطمینان هر دو روش با ۴ حالت مختلف

خروجی نهایی قابلیت اطمینان هر دو روش، در نمودار شکل ۱۰ نشان داده شده است.



شکل ۱۰) نمودار خروجی نهایی قابلیت اطمینان هر دو روش با ۴ حالت مختلف

۴. نتیجه‌گیری و کارهای آینده:

در نتایج به دست آمده از جداول با ۴ حالت مختلف، مشخص شد که با توجه به اعداد و ارقام ورودی، در هر دو روش پیشنهادی و روش مبتنی بر مارکوف، جواب‌های قابلیت اطمینان بهم نزدیک بودند و روش پیشنهادی نسبت به مارکوف پیشرفت بیشتر و بهتری دارد و حتی اختلاف حالت‌ها هم در هر دو روش بسیار کم هستند.

با توجه به مطالعات انجام شده و تحقیق ارائه شده و محدودیت‌های پژوهش، کارهای آتی پیشنهادی به شرح زیر می‌باشند: در روش‌های مطرح شده انتقال از یک مؤلفه به مؤلفه دیگر به صورت احتمالی در نظر گرفته شده است. در بسیاری از شرایط، احتیاج به بررسی وضعیت انتقال با قطعیت لازم است. به عنوان مثال ممکن است مؤلفه‌ای انتقال به صورت دنباله‌ای از چندین مؤلفه را داشته باشد، ولی می‌خواهد به صورت قطعی به مؤلفه بعدی برود لذا این موارد در این روش در نظر گرفته نشده است. بنابراین کار آینده در این زمینه می‌تواند وجود داشته باشد و همچنین توسعه و بسط روش پیشنهادی برای ارزیابی سبک‌های

چندریختی شامل دیگر سبک‌های تک‌ریختی و ارزیابی دیگر ویژگی‌های کیفی در سبک‌های چندریختی بسیار می‌تواند مورد توجه قرار گیرد و به وسیله ویژگی‌های کیفی دیگر در سبک‌های چندریختی باعث بهبود و کیفیت سیستم‌ها شود.

۵. منابع

- [۱] هاشمیان ح.۱۳۸۷. "مدلی برای بررسی ویژگی‌های کیفی در سبک‌های معماری چندریختی"، پایان‌نامه کارشناسی ارشد، دانشگاه آزاد اسلامی واحد علوم و تحقیقات.
- [2] I.Lakshmanan and S. Ramasamy." An artificial neural-network approach to software reliability growth modeling", 3rd International Conference on Recent Trends in Computing (ICRTC), 2015, vol. 57, pp. 695- 702
- [3] S. M. Sharafi, G. Aghaee Ghazvini and S. Emadi." An Analytical Model for Performance Evaluation of Software Architectural Styles", 2nd International Conference on Software Technology and Engineering (ICSTE), 2010, vol.1, pp. 394-398
- [4] G.ShahMohammadi." EVALUATION OF THE SOFTWARE ARCHITECTURE STYLES FROM MAINTAINABILITY VIEWPOINT", Computer Science & Information Technology (CS & IT), 2014, pp. 183–197
- [5] G. Shahmohammadi." Reliability Evaluation of Software Architecture Styles", Computer Science & Information Technology (CS & IT), 2016, pp. 117–129
- [6] Y.We, X.H.Shen1." Heterogeneous Architecture-Based Software Reliability Estimation: Case Study", International Conference on Convergence and Hybrid Information Technology, 2008, vol. 7, pp. 286- 290
- [7] W. Wang, D. Pan and M. Chen, "Architecture-based software reliability modeling", Journal of System and Software, 2006, vol. 79, pp. 132-146
- [8] Y.We, L. Wang and M. Wang." Software reliability analysis of Hierarchical architecture based on Markov model", Procedia Engineering, 2011, vol.15, pp.2857 – 2861
- [9] E. DUBROVA." FAULT TOLERANT DESIGN: AN INTRODUCTION", Department of Microelectronics and Information Technology, 2008, pp. 1 – 155
- [10] S.S.Jalali, H.Rashidi and E.Nazemi." A New Approach to Evaluate Performance of Component-based Software Architecture", UKSim 5th European Symposium on Computer Modeling and Simulation, 2011, pp. 451– 456
- [11] X. Wu, L. Huang Chen Li, L. Chen." Estimating Software System Reliability: An Architecture-Based Approach", 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2015, pp. 184- 191
- [12] G. Shahmohammadi." Reliability Evaluation of Software Architecture Styles", Computer Science & Information Technology (CS & IT), 2016, pp. 117–129
- [13] N. Eftekhari, M. Poyan Rad, H. Alinejad-Rokny."Evaluation and Classifying Software Architecture Styles Due to Quality Attributes", Australian Journal of Basic and Applied Sciences, 2011, pp.1251- 1256
- [14] Y.We, L. Wang, M. Wang." Software reliability analysis of Hierarchical architecture based on Markov model", Procedia Engineering, 2011, vol.15, pp.2857 – 2861
- [15] G.ShahMohammadi." EVALUATION OF THE SOFTWARE ARCHITECTURE STYLES FROM MAINTAINABILITY VIEWPOINT", Computer Science & Information Technology (CS & IT), 2014, pp. 183–197