فصلنامه علمی-ترویجی پدافند غیرعامل

سال سوم، شماره ۴، زمستان ۱۳۹۱، (پیاپی ۱۲): صص ۱-۱۰

اتکاپذیری در وب‌سرویس‌ها با رویکرد پدافند غیرعامل

محمدرضا حسنی‌آهنگر[۱]، مصطفی اخضمی[۲]

تاریخ دریافت: ۹۱/۰۹/۰۶
تاریخ پذیرش: ۹۱/۱۰/۰۴

چکیده

فناوری وب‌سرویس، روشی برای توسعه برنامه‌های توزیع‌شده با استفاده از واسط‌های ساده و تعریف‌شده، فراهم می‌کند. حصول اطمینان از اینکه وب‌سرویس‌ها قابل اعتماد هستند و می‌توانند درخواست‌های مشتریان خود را برآورده نمایند، به یک چالش برای وب‌سرویس‌ها تبدیل شده است. خرابی سرویس و یا بازگرداندن نتایج غلط در وب‌سرویس‌ها، ممکن است عواقب متعدد و بسیاری دربر داشته باشد. از این‌رو، اتکاپذیری وب‌سرویس‌ها به‌عنوان یکی از معیارهای پدافند غیرعامل، بسیار مهم تلقی می‌گردد. این مقاله در ابتدا به تعریف مختصری از اجزاء وب‌سرویس می‌پردازد؛ سپس طبقه‌بندی کلی از خطاهای وب‌سرویس و روش‌های مختلف تحمل‌پذیری خطا در آن‌ها بیان می‌شود. در انتها چند مورد از معماری‌های اتکاپذیری در وب‌سرویس‌ها از جمله معماری چندلایه برای تحمل‌پذیری نفوذ در وب‌سرویس و معماری FTWeb بررسی می‌شوند.

کلیدواژه‌ها: وب‌سرویس، اتکاپذیری، تحمل‌پذیری خطا، تکرارفعال، پدافند غیرعامل

۱- استادیار و عضو هیئت علمی دانشگاه جامع امام حسین(ع)

۲- دانشجوی کارشناسی ارشد مهندسی نرم‌افزار دانشگاه جامع امام حسین(ع) mostafa_akhzami@yahoo.com- نویسنده مسئول

۱- مقدمه

در سال‌های اخیر، فناوری‌ها و استانداردهای جدیدی برای توسعه نرم‌افزار ارائه شده است. وب‌سرویس، یک مدل برای سرویس‌های توزیع‌شده می‌باشد که از قابلیت دسترسی ساده و واسط‌های تعریف‌شده استفاده می‌کند. وب‌سرویس، نرم‌افزاری است که دسترسی به اطلاعات و سیستم‌های پردازش اطلاعات را به‌صورت توزیع‌شده فراهم می‌کند. سرویس‌های وب بر پایه SOAP[1]، WSDL[2]، XML و UDDI[3] استوارند. پروتکل SOAP وظیفه انتقال اطلاعات مبتنی بر XML را برعهده دارد. WSDL زبان توصیف سرویس‌وب، و UDDI محل ذخیره و دسترسی عمومی مشخصات وب‌سرویس است. یکی از چالش‌های پیش روی وب‌سرویس‌ها، اتکاپذیری آن‌ها است. از این‌رو، در مورد اتکاپذیری آن‌ها باید تدابیر مناسبی اندیشیده شود. از سوی دیگر، بسیاری از سیستم‌های اساسی و ملی کشور از وب‌سرویس برای ایجاد ارتباطات بین‌سیستمی استفاده می‌کنند. ارائه سرویس توسط این سیستم‌ها، حتی در صورت بروز خرابی، از نکات بسیار حائز اهمیت است. به همین دلیل، اتکاپذیری وب‌سرویس‌ها یک معیار مهم و اساسی در بحث پدافند غیرعامل می‌باشد. نفوذگران می‌توانند با نفوذ به این سرویس‌ها باعث ایجاد اختلال اساسی در آن‌ها شده و سیستم‌ها را از کار بیندازند. دو مسئله مهم در این‌جا مطرح است. یکی این‌که باید تا حد ممکن جلوی نفوذ به وب‌سرویس گرفته شود. دیگر این‌که اگر مهاجم بتواند به سیستم نفوذ پیدا کند و سیستم را دچار اختلال نماید، باید روش‌هایی برای تحمل‌پذیری خطا در سیستم پیش‌بینی نمود تا سیستم بتواند به کار خود ادامه دهد.

در این مقاله، یک تقسیم‌بندی از خطاهایی که در وب‌سرویس‌ها می‌تواند رخ دهد، ارائه شده است. سپس انواع روش‌های تحمل‌پذیری خطا در وب‌سرویس‌ها مطرح شده و چند مورد از این روش‌ها مانند تکرار فعال، غیرفعال و نیمه‌فعال مورد بررسی قرار گرفته است. در انتها به بررسی چند معماری اتکاپذیر، از جمله معماری FTWeb و معماری چندلایه برای تحمل‌پذیری نفوذ در وب‌سرویس‌ها پرداخته می‌شود.

۲- تعاریف و مفاهیم پایه

در این بخش، تعریف و اجزاء وب‌سرویس شرح داده می‌شود.

۲-۱- وب‌سرویس

بر طبق تعریف کنسرسیوم جهانی وب[4]، یک وب‌سرویس، نوعی سیستم نرم‌افزاری است که جهت تعامل ماشین با ماشین در سطح شبکه طراحی شده و دارای یک تعریف قابل پردازش توسط ماشین به نام WSDL است. دیگر سیستم‌ها بر طبق این توصیف، از قبل مهیا شده، با سرویس‌دهنده تعامل خواهند داشت و پیام‌های خود را تحت پروتکل SOAP منتقل می‌کنند. وب‌سرویس حاصل، ترکیب دو فناوری قدرتمند XML و HTTP می‌باشد[۱].

بنابراین تعریف، وب‌سرویس از اجزاء زیر تشکیل شده است.

۲-۱-۱- WSDL

زبان توصیف وب‌سرویس، زبانی مبتنی بر XML است که برای تعریف وب‌سرویس و توصیف چگونگی دسترسی به وب‌سرویس استفاده می‌شود. یکی از خواص وب‌سرویس‌ها، توصیف خود آن‌ها است. وب‌سرویس دارای اطلاعاتی است که نحوه استفاده از خود را توضیح می‌دهد. این توضیحات درWSDL نوشته می‌شود؛ متنی به زبان XML که به برنامه‌ها می‌گوید این وب‌سرویس چه اطلاعاتی به‌عنوان ورودی لازم دارد و چه اطلاعاتی را برمی‌گرداند[۱].

۲-۱-۲- SOAP

SOAP یک پروتکل سبک‌وزن برای تبادل اطلاعات در محیط‌های توزیع‌شده و غیرمتمرکز می‌باشد. این پروتکل مبتنی بر XML بوده و شامل سه بخش می‌باشد: ۱) پوششی که چارچوبی برای توصیف پیام بوده و چگونگی پردازش آن را تعریف می‌کند، ۲) مجموعه‌ای از قوانین رمزنگاری برای بیان نمونه‌هایی از انواع داده‌های تعریف‌شده ۳) قراردادی برای نمایش و فراخوانی و پاسخ از راه دور.

SOAP به‌طور بالقوه می‌تواند در ترکیب با انواع پروتکل‌های دیگر استفاده شود.SOAP دارای فرمت ویژه‌ای برای تبادل اطلاعات وب‌سرویس‌ها از طریق پروتکل HTTP می‌باشد. وقتی یک برنامه شروع به ارتباط با وب‌سرویس می‌کند، پیام‌هایSOAP وسیله‌ای برای ارتباط و انتقال دیتا بین آن‌دو می‌باشند. یک پیغام SOAPبه وب‌سرویس فرستاده می‌شود و یک تابع را در آن به اجرا درمی‌آورد. وب‌سرویس نیز از محتوای پیام SOAP استفاده کرده و عملیات خود را آغاز می‌کند. در انتها نیز نتایج را با یک پیام SOAP دیگر به برنامه اصلی می‌فرستد[۲].

۲-۱-۳- UDDI

UDDI استانداردی طراحی‌شده برای ارائه یک فهرست راهنمای قابل جستجو برای وب‌سرویس‌ها می‌باشد. بنابراین، مکان و موقعیت کارگزار سرویس را نمایش می‌دهد. در بسیاری موارد، UDDI مانند یک دفترچه تلفن طراحی شده است. شرکت‌ها می‌توانند وب‌سرویس خود را معرفی کنند، با وب‌سرویس دیگران آشنا شوند و آن را در سیستم‌های خود استفاده کنند[۳].

1- Simple Object Access Protocol
2- Web Services Description Language
3- Universal Description, Discovery and Integration
4- W3C

۳- طبقه‌بندی و مدیریت خطا در وب‌سرویس‌ها

خطای متفاوت، واکنشی متفاوت را می‌طلبد. می‌توان منابع خرابی‌ها را دسته‌بندی کرد و به‌وسیله این دسته‌بندی، عکس‌العمل مناسبی را اتخاذ نمود. هم‌چنین می‌توان در صورت مشاهده خرابی جدید، به دنبال عکس‌العمل مناسب‌تری گشت.

۳-۱- انواع خطا در وب‌سرویس‌ها

خطاهای وب‌سرویس را می‌توان در سه دسته اصلی بر اساس علت وقوع آن خطا طبقه‌بندی نمود.

۳-۱-۱- تخطی از توافق‌نامه سطوح خدمات

تخطی از توافق‌نامه سطوح خدمات و سیاست‌های وظیفه‌مندی (به‌عنوان مثال، محدودیت قیمت و یا تحویل فوری)، و نیازهای غیر وظیفه‌مندی (به‌عنوان مثال، زمان پاسخ سرویس، قابلیت دسترسی سرویس و امنیت). در این مورد، اجرای سرویس ممکن است به پایان برسد، امّا نتایج به‌دست آمده، منطبق بر مذاکرات توافق‌نامه سطوح خدمات و سیاست‌های همکاری نیستند[۴].

۳-۱-۲- خطاهای رفتاری و وظیفه‌مندی

ایجادکننده سرویس، نمی‌تواند اجرای وظیفه کامل یا ارائه خدمات را با توجه به اشتباهات محاسباتی و منطقی انجام دهد و باعث نتایج نادرست می‌شود. لذا جریان داده اشتباه یا ناسازگاری‌های معنایی در مبادله پیام‌ها را به‌دنبال دارد. خطاهای رفتاری می‌توانند در اثر مکالمات نامناسب، از قبیل فراخوانی نامناسب جهت بهره‌برداری سرویس و از دست دادن پیام‌ها در زمان پردازش ایجاد شوند[۴].

۳-۱-۳- خطاهای عملکردی

این خطاها شامل ناهنجاری‌های زیرساخت ارتباطات و خطاهای میان‌افزارها بر روی سرور و پایگاه داده میزبان می‌باشند. نمونه‌ای از این خطاها می‌تواند دردسترس نبودن شبکه که منجر به قطع‌شدن آن می‌شود، تراکم شبکه که باعث از دست رفتن پیام‌ها می‌شود، و یا سربار زیاد سرور که باعث تاخیر بیش از حد و وقفه‌های زیاد می‌شود، باشد[۴].

۳-۲- سیستم مدیریت خطا در وب‌سرویس‌ها

سیستم مدیریت خطا شامل ترکیبی از مراحل چندگانه نظارت[1]/ کشف[2]، تشخیص[3]، بازیابی[4]، راه‌اندازی مجدد[5] و تعمیر می‌باشد[۴].

1- Monitoring
2- Fault Detection
3- Diagnosis
4- Recovery
5- Restart

۳-۲-۱- نظارت/کشف خطا

نظارت و کشف خطا تشخیص می‌دهد که اتفاقی غیر منتظره رخ داده است. اجرای فرایند وب‌سرویس جهت پیدا کردن رفتارهای نادیده سیستم، با توجه به مدل رفتار عادی سیستم و ثبت اطلاعات کافی برای تشخیص‌های برخط و برون‌خط کنترل می‌شود[۴].

۳-۲-۲- تشخیص

تشخیص خطا، علت ریشه‌ای مشکل را در نقاطی که اقدامات اصلاحی را می‌توان انجام داد مشخص می‌کند. تشخیص خطا شامل هم تشخیص خطا و هم مکان خطا می‌باشد[۴].

۳-۲-۳- محدود کردن خطا

محدود کردن خطا، تلاش می‌کند تا تأثیر خطا را برای جلوگیری از گسترش اثرات خطا در یک ناحیه از وب‌سرویس محدود نماید. در نتیجه، از آلودگی مناطق دیگر وب‌سرویس جلوگیری می‌شود[۴].

۳-۲-۴- بازیابی

بازیابی، بهره‌گیری از روش‌هایی برای از بین بردن اثر خطا می‌باشد. سه روش بازیابی اصلی عبارت‌اند از: پوشش خطا، سعی دوباره و عقبگرد. تکنیک «پوشش خطا»، اثرات خطا را با اجازه جایگزینی اطلاعات به جای اطلاعات نادرست پنهان می‌کند. تکنیک «سعی دوباره»، متعهد به یک تلاش بیشتری در عملیات می‌باشد و بر این فرض استوار است که بسیاری از خطاها خاصیت گذرا بودن را دارند. تکنیک «عقبگرد» از عملیات وب‌سرویس نسخه پشتیبان (در نقاط خاص[6]) تهیه کرده و از آن در موارد خطا استفاده می‌کند. در برخی از این نقاط، ابتدا عملیات کشف خطا و سپس عملیات آغاز دوباره صورت می‌گیرد[۴].

۳-۲-۵- راه‌اندازی مجدد

راه‌اندازی مجدد، پس از آن‌که بازیابی اطلاعات صورت گرفت اتفاق می‌افتد. انواع راه‌اندازی مجدد عبارت‌اند از:

- راه‌اندازی آتشین[7]

از سرگیری تمام عملیات، از نقاطی که خطا تشخیص داده شده، صورت می‌گیرد.

- راه‌اندازی گرم[8]

تنها برخی از فرایندها را می‌توان بدون از دست رفتن به‌دست آورد.

6- Check Pointed
7- Hot Restart
8- Warm Restart

• راه‌اندازی سرد[1]

بارگذاری مجدد کامل سیستم بدون هیچ فرایندی. وب‌سرویس‌ها را می‌توان به وسیله راه‌اندازی مجدد سرور دوباره به‌کار انداخت[۴].

۴- تحمل‌پذیری خطا در وب‌سرویس‌ها

در این بخش، پارامترهای اتکاپذیری و روش‌های تحمل‌پذیری خطا در وب‌سرویس‌ها شرح داده می‌شوند.

۴-۱- پارامترهای اتکاپذیری

برای اطمینان از این‌که وب‌سرویس‌ها در لحظه‌ای که سرویس‌گیرنده به آن‌ها نیاز دارند در دسترس باشند، باید پارامترهای قابلیت اطمینان[2] و امنیت در طراحی سرویس‌ها و عناصر تشکیل‌دهنده آن‌ها در نظر گرفته شده باشد. در دسترس نبودن حتی یک قطعه از یک سرویس، دیگر اجزاء را تحت تأثیر خود قرار خواهد داد. دو پارامتر مهم و تأثیرگذار از اتکاپذیری در وب‌سرویس‌ها عبارت‌اند از:

۴-۱-۱- قابلیت دسترسی

خیلی از سیستم‌ها چنان طراحی شده‌اند که بدون وقفه و به صورت پیوسته عمل نمایند و سرویس مورد نظر را ارائه کنند. در بسیاری از شرایط، اهمیت دارد که در سیستم خرابی نبوده و تعداد خرابی هم مهم است. برای چنین سیستم‌هایی باید دانست چه کسری از زمان سیستم فعال است. این کار، قابلیت دسترسی[3] را مشخص می‌کند [۵].

۴-۱-۲- قابلیت اطمینان

قابلیت اطمینان یک سیستم R(t) در لحظه t عبارت است از احتمال اینکه سیستم در بازه زمانی[0,t] بدون خرابی باشد. در صورتی که در لحظه صفر، سیستم به صورت درست سرویس ارائه کرده باشد. قابلیت اطمینان، یک واحد اندازه‌گیری ممتد بدون ارائه سرویس صحیح سیستم می‌باشد[۵].

۴-۲- روش‌های مختلف تحمل‌پذیری خطا

تحمل‌پذیری خطا می‌تواند برای افزایش سطح قابلیت اطمینان، دسترس‌پذیری و سازگاری داده‌های وب‌سرویس مورد استفاده قرار گیرد. این فناوری شامل روش‌هایی مانند پیام‌های قابل اعتماد، تکرار[4]، نقاط بازرسی، ترمیم، ثبت پیام و تراکنش‌ها می‌باشد[۶].

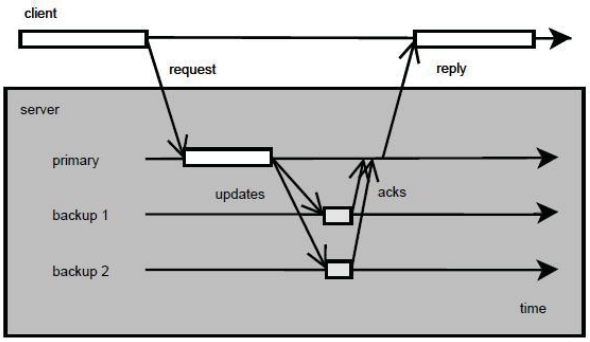
در اینجا تمرکز خود را روی روش تکرار به‌کار می‌بندیم.

۴-۲-۱- تکرار

تکرار در سیستم‌های تحمل‌پذیر خطا، برای حفاظت برنامه در مقابل خطا استفاده می‌شود. به‌طوری‌که اگر یک نسخه معیوب شود یکی دیگر از نسخه‌ها برای ارائه خدمت به سرویس‌گیرنده‌ها در دسترس باشد. انواع روش‌های تکرار، شامل تکرار فعال[5]، تکرار غیرفعال[6]، و تکرار نیمه‌فعال[7] می‌باشد[۶].

۴-۲-۱-۱- تکرار غیرفعال

در تکرار غیرفعال، تنها یک نسخه درخواست‌ها را دریافت، پردازش و پاسخ را تولید می‌کند. این نسخه، نسخه اولیه نامیده می‌شود و بقیه نسخه‌ها، نسخه‌های پشتیبان می‌باشند. مطابق شکل (۱)، نسخه اولیه، پیام‌های به‌روزشده (نقاط بازرسی[8] یا تغییر حالت) را زمانی که پردازش آن‌ها تمام شد، به نسخه‌های پشتیبان می‌فرستد. هنگامی که نسخه پشتیبان پیام را دریافت نمود آن را اعمال نموده و یک پیام بازگشت[9] به نسخه اولیه باز می‌گرداند. زمانی که نسخه اولیه، پیام بازگشتی را از هر نسخه پشتیبان دریافت نمود، پاسخ را به سرویس‌گیرنده می‌فرستد. اگر نسخه اولیه با خطا مواجه شود، نسخه‌های پشتیبان از سرویس عضویت در گروه، برای انتخاب یک نسخه اولیه دیگر استفاده می‌کنند. نسخه اولیه از آخرین نقطه بازرسی، بازیابی می‌شود و کار پردازش درخواست‌ها را ادامه می‌دهد. اگر نسخه پشتیبان با خطا مواجه شود، گروه سرور، آن را از پروتکل گروه خارج می‌نماید. خطا در نسخه اولیه باعث افزایش زمان پاسخگویی می‌شود[۷].



شکل ۱- تکرار غیرفعال [۷]

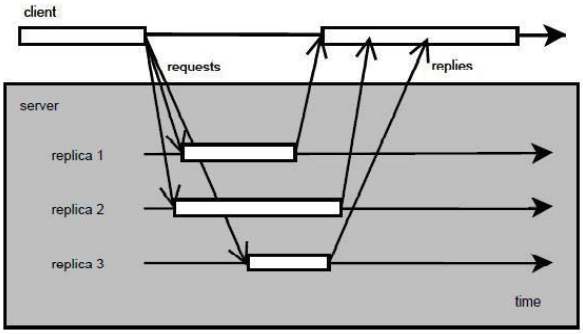
۴-۲-۱-۲- تکرار فعال

در تکرار فعال همه نسخه‌ها کار یکسانی را انجام می‌دهند. هنگامی که یک درخواست رسید، به‌وسیله تمام نسخه‌ها دریافت و پردازش

1- Cold Restart
2- Reliability
3- Availability
4- Replication

5- Rctive Replication
6- Passive Replication
7- Semi-Active Replication
8- Checkpoints
9- ACK

می‌شود و هر نسخه یک پاسخ صحیح را ارائه می‌نماید. سرویس‌گیرنده منتظر می‌ماند تا اولین پاسخ را دریافت کند. این رفتار در شکل (۲) نشان داده شده است. این روش، افزونگی سریع در موردی که یکی از نسخه‌ها در اجرا با خطا مواجه شود، را درپی دارد. سرویس‌گیرنده درخواستی را می‌فرستد و یک پاسخ از نسخه دیگری که خطا ندارد، بدون تاخیر زیاد دریافت می‌کند. این کار منجر به زمان پاسخ کوتاه - حتی در صورت وجود خطا - می‌شود[۷].



شکل ۲- تکرار فعال [۷]

۴-۲-۱-۳- تکرار نیمه‌فعال

این روش، ترکیبی از دو اجرای غیرقطعی[1] و تکرار فعال می‌باشد. یکی از نسخه‌ها سردسته[2] نامیده می‌شود و دیگر نسخه‌ها پیروان[3] آن می‌باشند. مانند تکرار فعال، همه نسخه‌ها درخواست را پردازش می‌کنند؛ اما سردسته قطعات غیرقطعی را پردازش و به پیروان اطلاع می‌دهد. سردسته تصمیم می‌گیرد که درخواست سفارشی باید به‌کار گرفته شود و تصمیمات خود را به پیروان می‌فرستد. هم‌چنین تنها، پاسخ سردسته به‌وسیله سرویس‌گیرنده دریافت می‌شود، زیرا پاسخ‌های یکسان توسط سیستم ارتباطی حذف می‌شوند[۷].

۵- معماری‌های اتکاپذیر وب‌سرویس

در این بخش به بررسی دو نمونه از معماری‌های وب‌سرویس اتکاپذیر شامل معماری چندلایه برای تحمل‌پذیری نفوذ در وب‌سرویس و معماری تحمل‌پذیری خطا برای وب‌سرویس‌ها[4] پرداخته می‌شود.

۵-۱- معماری چندلایه برای تحمل‌پذیری نفوذ در وب‌سرویس‌ها

تکنیک‌های تحمل‌پذیر خطا[5] در نرم‌افزار به دو دسته اصلی تکنیک‌های تک‌نسخه‌ای و تکنیک‌های چندنسخه‌ای تقسیم می‌شوند. انتخاب بهترین روش طراحی، بستگی به مشخصات نرم‌افزار و نیازمندی‌های امنیتی، از جمله قابلیت دسترسی، جامعیت و محرمانگی دارد. در این معماری از مدل تک‌نسخه‌ای استفاده شده است.

۵-۱-۱- اجزاء معماری

در این معماری، چهار بنای اصلی وجود دارد که لایه‌های مختلفی از امنیت در بخش‌های کلیدی سیستم به‌منظور شناسایی، پیشگیری، محدود نمودن و تحمل حملات را شامل می‌شوند. این معماری از یک طرح اولیه و پشتیبانی ساده استفاده می‌کند. سرویس‌های اولیه و پشتیبان، پیاده‌سازی یکسانی داشته و در نتیجه، آسیب‌پذیری‌های مشابهی دارند. امّا اگر بتوان از حملاتی که قبلاً شناسایی شده‌اند، به‌طور مداوم بر روی سیستم جلوگیری کرد، آن‌گاه می‌توان از مزایای تکنیک تحمل‌پذیر خطای تک‌نسخه‌ای حتی در مورد خطاهای مخرب استفاده نمود[۸].

۵-۱-۲- ابزارهای تشخیص و پیشگیری

در این‌جا به بررسی ابزارهای تشخیص و پیشگیری در معماری چندلایه برای تحمل‌پذیری نفوذ در وب‌سرویس‌ها می‌پردازیم.

۵-۱-۲-۱- دیواره‌آتش وب‌سرویس

این برنامه، دومین لایه دفاعی در این معماری را شامل می‌شود. با دیواره آتش وب‌سرویس، می‌توان قبل از ارسال پیام‌ها به وب‌سرویس، اعتبار آن‌ها را با الگوی XML سخت‌گیرانه تأیید نمود و در برابر درخواست‌های نامعتبر و یا مخرب، وب‌سرویس را محافظت کرد. تعیین محدوده بالا و پایین برای تعدادی از عناصر در اسناد، محدود نمودن طول ورودی، تعیین کاراکترهای معتبر، اندازه کل درخواست‌ها و هر چیزی که ممکن است توسط مهاجم مورد سوء استفاده قرار گیرد، توسط الگوی XML سخت‌گیرانه می‌تواند مورد استفاده قرار گیرد[۸].

۵-۱-۲-۲- تشخیص نفوذ

استفاده از ترکیبی از دو شیوه غیرمتعارف و شناخته شده، مکانیزم تشخیص سوء استفاده‌ها، و رفتارهای غیرطبیعی از اجزاء این معماری می‌باشند. پوشش نفوذ[6] به‌طور معمول با تکرار به‌دست می‌آید. بنابراین قابل اجرا در تکنیک تک‌نسخه‌ای یا معماری افزونگی با توجه به این‌که همان ورودی را به خروجی یکسان در همه نسخه‌ها کپی می‌نماید، نمی‌باشد. در این سیستم، الگوهای فعالیت، با الگوهای فعالیت نرمال و غیر نرمال برای تشخیص سوء استفاده و یا رفتارهای

1- Non-Deterministic
2- Leader
3- Followers
4- FTWeb
5- Fault Tolerance
6- Intrusion Masking

غیرطبیعی مقایسه می‌شود. دو مکانیزم تشخیص در این معماری، شامل استفاده از تست پذیرش[1] برای درخواست کنترل اعتبار بر روی پاسخ، و حالت مکانیزم انتقال برای تعیین حالت پذیرفته شده بعدی می‌باشند. سیستم تشخیص نفوذ در این معماری، هم‌چنین بازرسی‌های برون‌خط بر روی حفاظت فایل‌های ثبت وقایع[2]، جهت تشخیص فعالیت‌های غیر طبیعی را انجام می‌دهد[۸].

۵-۱-۲-۳- محدود نمودن نفوذ

برای ایجاد سیستم‌های بحرانی تحمل‌پذیر خطا، باید گسترش وقوع خطا را به‌طور خودکار به محض این‌که تشخیص داده شد، محدود نمود. این مؤلفه تضمین می‌کند که خسارات مستقیم یا غیر مستقیم ناشی از درخواست‌های مخرب محدود شوند[۸].

۵-۱-۲-۴- بازیابی و پیکربندی مجدد

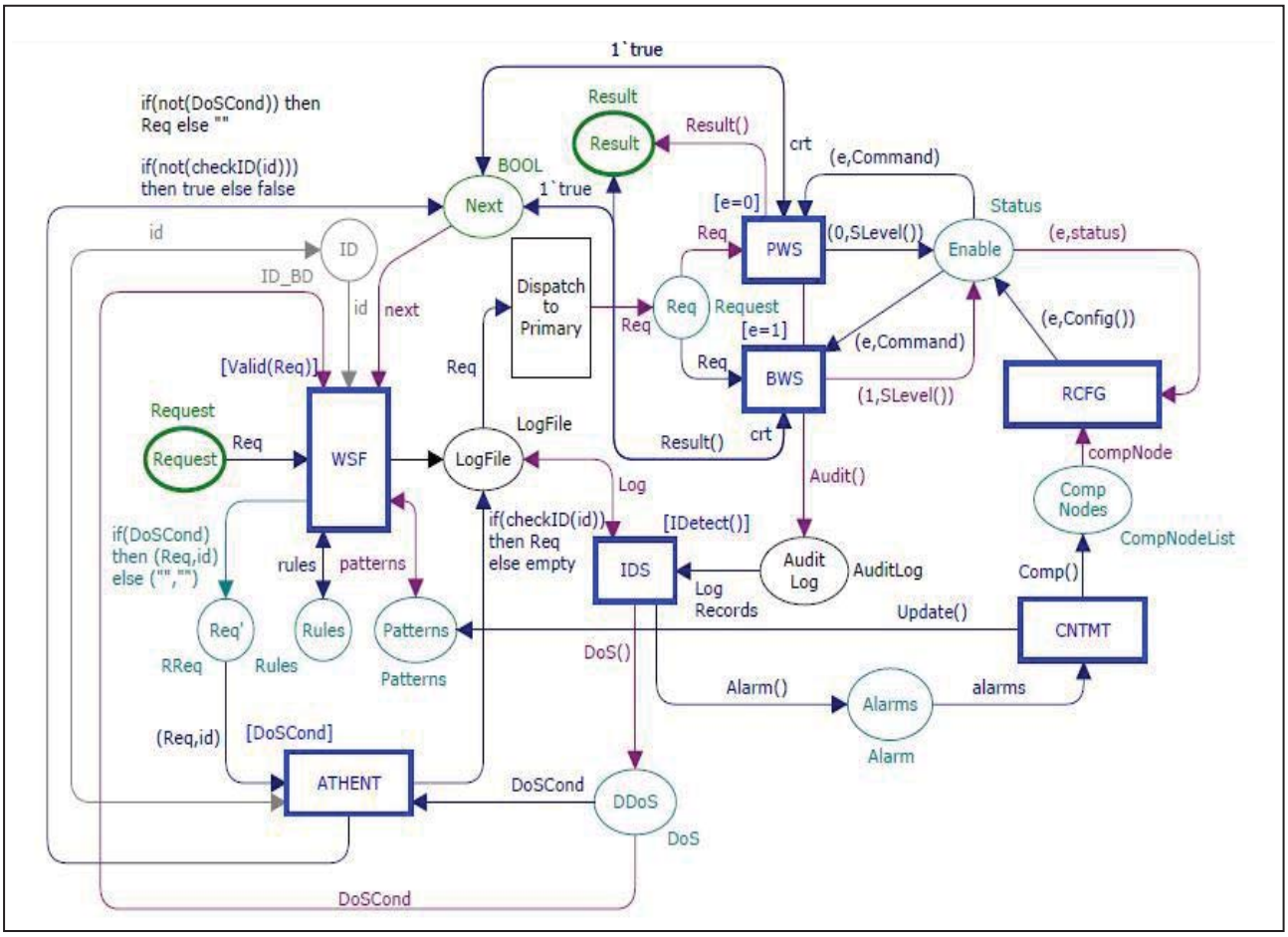
مکانیزم‌های بازیابی در این معماری عبارت‌اند از:

• بازیابی نفوذ

مکانیزم بازیابی و بازرسی رویداد امنیتی، از دسترسی‌های غیر مجاز به منابع سیستم که تلاش می‌کنند سیاست‌های جامعیت سیستم را نقض کنند، جلوگیری می‌کند. هر دسترسی به فایل‌های وب‌سرویس‌ها توسط یک فرایند پیچیده باید تایید شود[۸].

• بازیابی خطا

سرویس‌های معیوب به‌وسیله سیگنال‌های دوره‌ای و تصادفی با مجموعه‌ای ازپیش‌تعریف‌شده از "درخواست و پاسخ" که توسط مدیر پیکربندی فرستاده می‌شود، شناسایی می‌شوند. با مقایسه پاسخ‌های سرویس با پاسخ مورد نظر، می‌توان سرویسی که به خطر افتاده را تشخیص داد. هنگامی که خرابی در طرح اولیه تشخیص داده شد، ارائه سرویس به کاربر نهایی از طریق جایگزینی طرح پشتیبان با طرح اولیه ادامه می‌یابد[۸].



شکل ۳- مدل شبکه پتری معماری چندلایه [۸]

1- Acceptance Test 2- Log Files

۵-۱-۳- مدل شبکه پتری رنگی

در شکل (۳) مدل شبکه پتری رنگی این معماری نشان داده شده است. ابتدا قسمت WSF درخواست کاربر را دریافت نموده و آن را با استفاده از الگوهای شناخته‌شده حمله و قوانین تصدیق[1] تأیید می‌نماید. پس از مرحله تصدیق، درخواست وارد شده در سیستم ثبت و به وب‌سرویس اولیه فرستاده می‌شود. حالت گذر RCFG سطح سرویس را در وب‌سرویس اصلی تأیید می‌نماید. اگر سطح سرویس قانع‌کننده نبود آن‌گاه نسخه پشتیبان به جای نسخه اصلی استفاده می‌شود. حالت گذر IDS فایل ثبت وقایع را برای شناسایی و هشدار دادن جهت رفتارهای غیرطبیعی بررسی می‌کند. با اعلام هرگونه هشدار، حالت گذر CNTMT جهت تهیه لیستی از گره‌هایی که با خطر مواجه شده‌اند، به‌کار گرفته شده و گذر حالت RCFG تحریک شده و سرویس‌های به خطر افتاده در وب‌سرویس اصلی را غیر فعال می‌سازد. اگر سیستم وارد شرایط حمله DDOS شده باشد، حالت گذر ATHENT فعال شده و درخواست‌ها را دریافت نموده و صحت ID اختصاص داده شده را تأیید می‌کند. هنگامی که سیستم آمادگی پاسخ به درخواست‌های پذیرفته شده را دارد، حالت بعدی به طور صحیح جایگزین شده و درخواست بعدی به سیستم وارد می‌شود[۸].

۵-۲- معماری تحمل‌پذیری خطا برای وب‌سرویس‌ها[2]

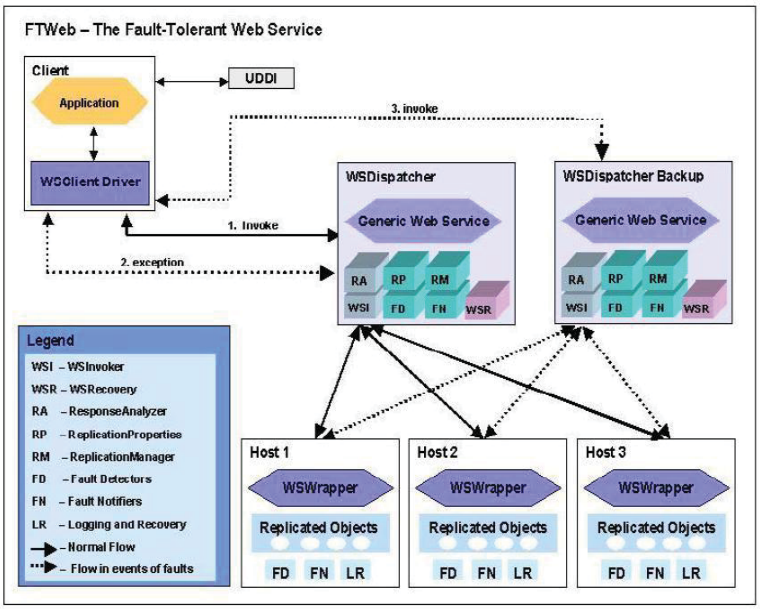
تکسیرا سانتوز[3] و همکارانش این معماری را در سال ۲۰۰۵ ارائه کردند. این معماری از تکرار فعال برای تحمل‌پذیری خطا استفاده می‌کند[۹].

۵-۲-۱- ساختار FTWeb

ایده اساسی معماری FTWeb، استقرار روش تکرار فعال برای دستیابی به تحمل‌پذیری خطا در معماری سرویس‌گرا می‌باشد. نسخه‌های مختلف برای یک سرویس معین در یک گروه سازمان‌دهی می‌شوند و همه نسخه‌ها درخواست را دریافت، اجرا و پاسخ را به کاربر ارائه می‌دهند. این روش اجازه می‌دهد تا اشیاء توزیع‌شده بر روی سرورهای پراکنده (در حوزه‌های مختلف) تکرار شده و مدیریت خود را به زیرساخت FTWeb واگذار نمایند. مطابق شکل (۴)، اجزاء این معماری عبارت‌اند از:

۵-۲-۲- WSClient Driver

مؤلفه WSClient Driver مسئولیت تشخیص خطای بخش موتور توزیع‌کننده وب‌سرویس[4] و انتقال درخواست به آن را بر عهده دارد. پشتیبان‌گیر نیز بر روی یک سرور مستقل قرار دارد. این جزء به‌عنوان یک رهگیر در لایه SOAP تعریف شده و بر روی سرویس‌گیرنده قرار گرفته است. هدف این معماری این است که یک تحمل خطای شفاف به کاربران برنامه‌ها ارائه نماید. در نتیجه، موتور توزیع‌کننده وب‌سرویس از تبدیل شدن به یک نقطه بحرانی خطا جلوگیری می‌نماید. موتور توزیع‌کننده وب‌سرویس باید پس از خرابی درخواست را پردازش کند؛ برای این‌که در هنگام پاسخ به مشتری، اجزاء WSClient Driver درخواست را به موتور پشتیبان WSClient Driver انتقال می‌دهد، که سرویس تکرارشده را فراخوانی نماید.



شکل ۴- ساختار FTWeb [۹]

1- Validation
2- A Fault Tolerant Infrastructure for Web Services (FTWeb)
3- Teixeira Santos
4- WSDispatcher

از طریق مکانیزم ثبت موجود در نسخه‌ها، بررسی می‌شود که آیا درخواست در حال حاضر پردازش شده و سپس نسخه به سادگی، پاسخ را به موتور پشتیبان توزیع‌کننده وب‌سرویس برمی‌گرداند. شکل (۴)، هر دو جریان طبیعی و جریان رخداد خطا در موتور توزیع‌کننده وب‌سرویس را نشان می‌دهد[۹].

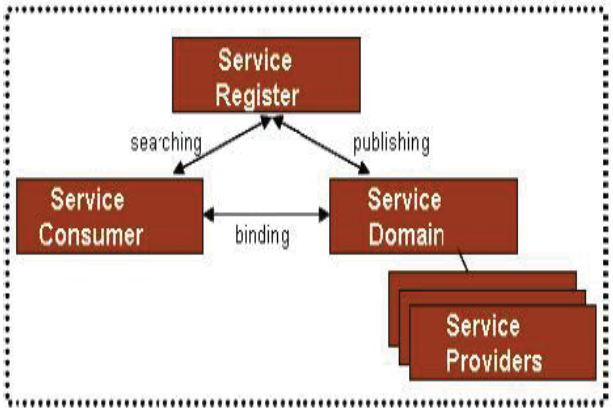
۵-۲-۳- موتور توزیع‌کننده وب‌سرویس

توزیع‌کننده وب‌سرویس، جزء اصلی معماری FTWeb می‌باشد و نیز مسئول مکانیزم‌هایی برای مدیریت نسخه‌ها، فراخوانی به صورت همزمان، خدمات نسخه‌ها، تجزیه‌وتحلیل پاسخ‌های پردازش‌شده، تشخیص و شروع فرایند بازیابی برای نسخه‌های معیوب می‌باشد. توزیع‌کننده وب‌سرویس از اجزاء زیر تشکیل شده است[۹]:

۵-۲-۳-۱- وب‌سرویس عمومی

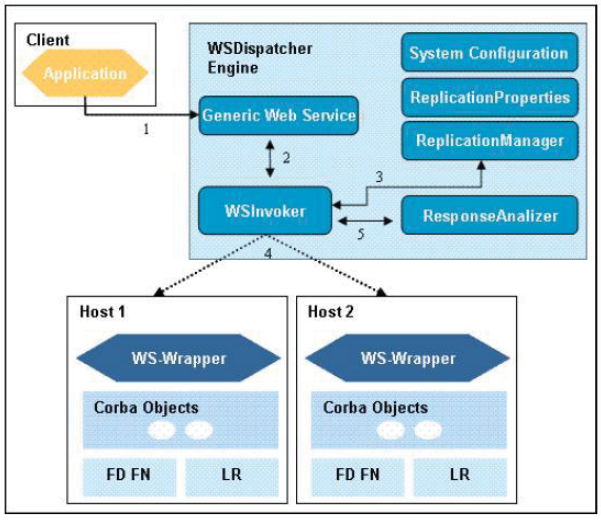
وب‌سرویس عمومی مسئول دستیابی سرویس‌گیرنده در ارجاع به وب‌سرویس و پارامترهای مورد نیاز برای اجرای آن می‌باشد. پس از اجرا، این مؤلفه مسئول بازگرداندن پاسخ به سرویس‌گیرنده می‌باشد. استفاده از این مؤلفه باعث می‌شود تا سرویس‌گیرنده مجموعه‌ای از نسخه‌ها که مستقل و از لحاظ جغرافیایی پراکنده هستند را به‌عنوان یک سرویس واحد مشاهده نماید.

به منظور ایجاد گروه‌ها در روش تکرار، لازم است مفهوم دامنه سرویس استفاده شود. یک دامنه سرویس اجازه تجمیع و به‌اشتراک‌گذاری توصیف سرویس‌های چندگانه را می‌دهد. انقیاد اطلاعات ارجاع‌شده به گروه، اجازه می‌دهد چندین سرویس به‌عنوان تنها یک سرویس مجازی واحد در نظر گرفته شوند. قوانینی را می‌توان برای دامنه جهت کنترل رفتار سرویس‌های مجتمع به‌کار بست. شکل (۵)، تفاوت بین مدل دامنه سرویس و مدل وب‌سرویس معمولی را نشان می‌دهد[۹].



شکل ۵- دامنه سرویس[۹]

موتور توزیع‌کننده وب‌سرویس شامل یک پیکربندی سیستم می‌باشد که در آن مدیر سرویس، گروه را ایجاد و از طریق WSDL اسناد نسخه‌ای که بخشی از این گروه است را نشان می‌دهد. در این سیستم، تکرار و خصوصیات مدیر خطا تعریف شده است[۹].



شکل ۶- عملیات WSInvoker [۹]

۵-۲-۳-۲-WSInvoker

عملکرد این مؤلفه و تعامل با سایر مؤلفه‌ها مطابق شکل (۶) می‌تواند توسط یک توالی پنج مرحله‌ای شرح داده شود[۹].

- سرویس‌گیرنده، وب‌سرویس عمومی را فراخوانی می‌کند. آن‌گاه اجرا و پارامترهای مورد نیاز برای فراخوانی به وب‌سرویس عمومی داده می‌شود.
- مؤلفه وب‌سرویس عمومی، WSInvoker را فراخوانی و اطلاعات به دست آمده از سرویس‌گیرنده را به آن انتقال می‌دهد.
- WSInvoker با مدیر تکرار، همکاری کرده و خواص مؤلفه‌های تکرار را برای به‌دست آوردن محل نسخه‌ها و خواص تحمل‌پذیری خطا برای این گروه معین می‌کند.
- WSInvoker سرویس تکرار را از حوزه‌های مختلف فراخوانی کرده و اجرای آن‌ها را مدیریت می‌کند.
- پس از اخذ پاسخ از همه نسخه‌ها، WSInvoker مؤلفه تحلیل‌گر پاسخ[1] را فراخوانی می‌کند. تحلیل‌گر پاسخ در میان پاسخ‌ها، کار رأی‌گیری را انجام می‌دهد. سپس WSInvoker پاسخی را که توسط تحلیل‌گر پاسخ آماده شده به وب‌سرویس عمومی ارسال می‌کند[۹].

اگر یک نسخه در لحظه‌ای که اجرا می‌شود یک خطا را ایجاد کند یا پاسخ در یک محدوده زمانی داده نشود، WSInvoker مکانیزم‌های اطلاع‌رسانی را فعال می‌کند؛ به‌گونه‌ای که مدیر تکرار می‌تواند نسخه

1- Response Analyzer

معیوب را از گروه حذف نماید. نسخه معیوب تا زمانی که از طریق مکانیزم‌های بازیابی دوباره به‌طور صحیح قادر به سرویس‌دهی باشد، از این گروه اخراج می‌شود[۹].

۵-۲-۳-۳- تحلیل‌گر پاسخ

مؤلفه تحلیل‌گر پاسخ اختیاری می‌باشد و به‌عنوان یک رای‌دهنده عمل می‌کند. بعد از آن‌که نسخه اجرا شد، مؤلفه WSInvoker پاسخ را به تحلیل‌گر پاسخ واگذار می‌کند. سپس تحلیل‌گر همه پاسخ‌های به‌دست آمده را تجزیه و تحلیل می‌کند. پاسخی که بیشترین تعداد تکرار را دارد، انتخاب می‌شود. این مؤلفه می‌تواند در تحمل‌پذیری خطا استفاده شود[۹].

۵-۲-۳-۴- مدیر تکرار

مؤلفه مدیر تکرار[1] ویژگی‌های مدیریت نسخه‌های FT-CORBA را گسترش می‌دهد. این مؤلفه به‌صورت پویا، اضافه نمودن نسخه جدید و حذف نسخه‌های معیوب را با توجه به قوانین تعریف شده در خصوصیات تکرار، کنترل می‌کند[۹].

۵-۲-۳-۵- خصوصیات تکرار

این مؤلفه عمل نگاشت خصوصیات تحمل‌پذیر خطای تعریف‌شده در FT-CORBA را برای ساختار FTWeb انجام می‌دهد. موتور توزیع‌کننده وب‌سرویس با توجه به ویژگی‌های پیکربندی سیستم به مدیریت سرویس اجازه می‌دهد تکرار و خصوصیات مدیریت خطا را تعریف نماید. این خصوصیات عبارت‌اند از:

- **شیوه تکرار**

سبک و شیوه‌ای از تکرار که شامل تکرار غیرفعال سرد، تکرار غیرفعال گرم و تکرار فعال می‌باشد را تعیین می‌کند.

- **شیوه نظارت**

شیوه‌ای از نظارت بین کشش[2] و فشار[3] را تعیین می‌کند. در روش کشش، ردیاب خطا به صورت دوره‌ای پیام‌هایی را برای نسخه‌هایی که مورد بررسی هستند ارسال می‌کند، که ببیند آیا فعال هستند یا خیر. در روش فشار، نسخه‌ها به‌صورت دوره‌ای پیام‌هایی را به ردیاب خطا ارسال می‌کنند. با این کار نشان می‌دهند که نسخه فعال می‌باشد.

- **بازه زمانی نظارت و اتمام مهلت**

بازه زمانی نظارت[4] و حداکثر زمان پاسخ[5] برای سرویسی که تحت نظارت است، تعیین می‌کند که آن سرویس معیوب است یا خیر.

- **اتمام مهلت پاسخ[6]**

محدودیت زمانی پاسخ برای سرویسی که توسط WSInvoker فراخوانی شده، را تعیین می‌کند.

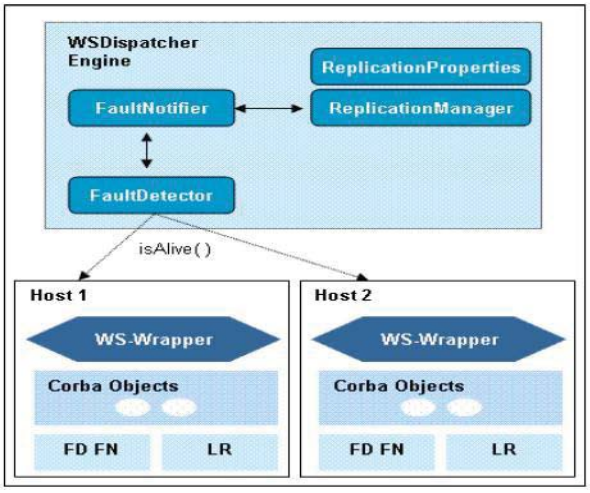
- **بازیابی**

نمایان‌گر فرایند بازیابی سرویس‌ها می‌باشد. حالت سرویس می‌تواند به‌طور خودکار با استفاده از مکانیزم‌های ارائه‌شده توسط معماری FTWeb یا به‌صورت دستی توسط مدیر بازیابی شود[۹].

۵-۲-۳-۶- آشکارساز خطا

آشکارساز خطا[7]، تشخیص خطا و ویژگی‌های اطلاع‌رسانی FT-CORBA را به وب‌سرویس گسترش می‌دهد. برای یک وب‌سرویس که باید مورد نظارت قرار گیرد، با استفاده از پیاده‌سازی رابط PullMonitorable متد حاوی isAlive() فراخوانی می‌شود. با فراخوانی این متد، مؤلفه ردیاب خطا بر روی نسخه‌ها عمل نظارت را انجام می‌دهد. نظارت با توجه به ویژگی‌های به‌دست‌آمده از طریق خصوصیات تکرار و تعریف پیکربندی سیستم، انجام می‌شود[۹].

زمانی که یک خطا رخ می‌دهد، مؤلفه آشکارساز خطا، یک اخطار را از آشکارساز خطا دریافت می‌کند. اخطاردهنده خطا به مدیر تکرار اطلاع می‌دهد که نسخه‌های معیوب را از گروه وب‌سرویس حذف نماید. شکل (۷) مدیریت خطا در معماری FTWeb را نشان می‌دهد[۹].



شکل ۷- مدیریت خطا در FTWeb [۹]

1- Replication Manager
2- PULL
3- PUSH
4- Ping
5- Timeout

6- Response Time Out
7- Fault Detector

۵-۲-۳-۷- WSRecovery

این مؤلفه مسئول بازیابی نسخه‌های معیوب می‌باشد. موتور توزیع‌کننده وب‌سرویس یک کنسول نظارت[1] دارد، که همه نسخه‌هایی که در طول درخواست برای یک تراکنش یا در طول فرایند نظارت با خطا مواجه می‌شوند را نمایش می‌دهد. این کنسول اجازه می‌دهد تا مدیر سرویس، فرایند بازیابی برای یک یا تعدادی از نسخه‌ها را آغاز نماید. مدیر می‌تواند حالت سرویس در صورت بروز یک خطا در همه نسخه‌ها را اطلاع دهد، یا فرایند بازیابی را فقط برای یک نسخه معیوب آغاز نماید. این فرایند به‌عنوان بازیابی دستی شناخته می‌شود[۹].

در بازیابی خودکار، WSRecovery به‌صورت دوره‌ای نسخه‌های معیوب را چک کرده، حالت نسخه‌های غیر معیوب را به‌دست آورده و از طریق مکانیزم رأی‌گیری به‌وسیله مؤلفه تحلیل‌گر پاسخ، حالت نسخه معیوب را دوباره بازسازی می‌کند. عملیات این مؤلفه شبیه WSInvoker می‌باشد. این قابلیت یا از طریق کنسول مدیریت و یا از طریق اعلام اخطار توسط آشکارساز خطا (زمانی که یک نسخه معیوب را تشخیص می‌دهد) فراخوانی می‌شود[۹].

۵-۲-۴- WSWrapper

به‌منظور یکپارچگی بین CORBA و فناوری وب‌سرویس، یک مؤلفه WSWrapper برای ایجاد روابط بین موتور توزیع‌کننده وب‌سرویس و اشیائی که توسط فرایند درخواست کاربر آماده می‌شود، ایجاد شده است. از طریق این مؤلفه، درخواست‌های SOAP به شیء CORBA تبدیل می‌شود. WSWrapper از رابط فراخوانی خودکار برای فراخوانی اشیاء استفاده می‌کند و می‌تواند در اجرای هر شیء CORBA بر روی ارائه‌دهنده سرویس استفاده شود. از طریق این روش، تکرار اشیاء روی سرورهای پراکنده جغرافیایی امکان‌پذیر شده و مدیریت آن‌ها به موتور توزیع‌کننده وب‌سرویس واگذار می‌شود[۹].

۶- نتیجه

وب‌سرویس، یک فناوری جدید برای یکپارچه‌سازی فرآیندهای کسب‌وکار می‌باشد. وب‌سرویس‌ها برای رسیدن به هدف کسب‌وکار در سازمان‌های مختلف باید اتکاپذیر باشند. یکی از مهم‌ترین چالش‌ها در وب‌سرویس‌ها ارائه خدمات به‌صورت درست و دسترس‌پذیری آن‌ها با وجود خطا است. در این مقاله، یک طبقه‌بندی از خطاهایی که ممکن است در وب‌سرویس‌ها رخ دهد، بیان شد. هم‌چنین انواع روش‌های تحمل‌پذیری خطا در وب‌سرویس‌ها، از جمله تکرار فعال، تکرار غیرفعال و تکرار نیمه‌فعال ارائه گردید. در انتها چند نمونه از معماری‌های اتکاپذیر وب‌سرویس‌ها، از جمله معماری چندلایه برای تحمل‌پذیری نفوذ در وب‌سرویس و معماری FTWeb شرح داده شد و روش‌ها و متدهایی که این معماری‌ها از آن استفاده می‌نمایند به‌طور کامل آمده است. نتیجه نهایی این‌که وب‌سرویس‌ها باید دارای مکانیزم‌هایی برای بالابردن سطوح قابلیت اطمینان، دسترس‌پذیری و سازگاری را در خود ایجاد نمایند. بر این اساس، مباحث تحمل‌پذیری خطا جهت ایجاد سیستم اتکاپذیر در وب‌سرویس‌ها به‌عنوان یکی از موضوعات اساسی پژوهش‌ها، جهت تحقق مباحث پدافند غیرعامل مطرح خواهد بود.

1- Monitoring Console

مراجع

1. Socrates Krishnamurthy, Charles Stevens, Ramnath Nair; "An Overview Of Web Services"; University of Illinois – Springfield; (**2011**).
2. http://www.w3.org/TR/SOAP;"Simple Object Access Protocol (SOAP)"; (**2000**).
3. Gunzer,Hartwig;"Introduction to Web Services"; (**2002**).
4. HN, Lakshmi, Mohanty, Hrushikesha; "Automata for Web Services Fault Monitoring and Diagnosis"; Special Issue of IJCCT, Volume 3, Issue-2,pp 13-18; (**2010**).
5. DUBROVA,ELENA, "FAULT TOLERANT DESIGN: AN INTRODUCTION"; Department of Microelectronics and Information Technology Royal Institute of Technology Stockholm, Sweden; Kluwer Academic Publishers; (**2008**).
6. Moser, L. E., Melliar-Smith, P. M., Zhao, Wenbing; "Making Web Services Dependable"; Electrical and Computer Engineering University of California, Santa Barbara; (**2005**).
7. Kolltveit,Heine; "Techniques for Achieving Exactly-Once Execution Semantics and High Availability for Multi-Tier Applications"; Norwegian University of Science and Technology Department of Computer and Information Science; (**2004**).
8. Aghajani, Zahra, AbdollahiAzgomi,Mohammad; "A Multi-Layer Architecture for Intrusion Tolerant Web Services"; International Journal of u- and e- Service, Science and Technology; p73-80; (**2009**).
9. Teixeira Santos, Giuliana, Lau Cheuk Lung, Montez, Carlos, "FTWeb:A Fault Tolerant Infrastructure for Web Services"; Proceedings of the 2005 Ninth IEEE International EDOC Enterprise Computing Conference; (**2005**).

The Dependability of Web Services with a Passive Defense Approach

M. R. Hasani Ahangar[1]

M. Akhzami[2]

Abstract

Web services technology provide a method for developing distributed applications by using simple and defined interfaces. However, ensuring that Web services are reliable, and are able to meet the demands of its customers, has become a challenge for Web services. Service failure or restoring false results in Web services, may have numerous consequences. Therefore dependability of Web services in passive defense is considered as very important. In this article ,a brief definition of components web services is given. Then a general classification of fault in Web Services and different methods of fault tolerance in Web services is provided. Finally, a few of the dependability of web services architectures including a multi-layer architecture for intrusion tolerant web services architecture and FTWeb architecture are also investigated.

Key Words: *Web Service, Dependability, Fault Tolerance, Active Replication, Passive Defense*

1- Imam Hossein Comprehensive University, Assistant Professor and Academic Member
2- Imam Hossein Comprehensive University, M.S in Software Engineering (mostafa_ akhzami@yahoo.com) - Writer in Charge