

УДК 004.415.5

А.И. МИХАЙЛИЧЕНКО, А.В. ГОРБЕНКО, В.С. ХАРЧЕНКО

*Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина***МОНИТОРИНГ И ОБРАБОТКА ИСКЛЮЧЕНИЙ
ДИНАМИЧЕСКИ РЕКОНФИГУРИРУЕМЫХ WEB-СЛУЖБ**

Проведен анализ факторов, влияющих на готовность и функциональную надежность SOA; построена модель поведения MiddleWare при возникновении исключений; приведены результаты экспериментальных исследований мониторинга Web-сервисов.

Web-сервисы, качество обслуживания, готовность, обработка исключений**Введение**

При использовании Сервис-Ориентированной Архитектуры (Service Oriented Architecture, SOA) предполагается, что система состоит из совокупности слабосвязанных Web-сервисов (WS) – приложений, предоставляющих набор определенных услуг и взаимодействующих посредством Internet-протоколов. Web-сервисы могут взаимодействовать друг с другом и предоставлять, таким образом, сложные композитные услуги. [1, 2]. С точки зрения разработки программного обеспечения, SOA и WS представляют сдвиг парадигмы в процессе разработки программного обеспечения от покомпонентной к объединенной валидации и верификации, включая спецификацию, дизайн, реализацию, тестирование, моделирование и оценку надежности [1].

Современная концепция построения гарантоспособных сервис-ориентированных систем предусматривает выполнение входного и выходного контроля, для гарантии того, что клиенты получают услуги, заслуживающие доверия.

Для реализации этой концепции была предложена идея использования промежуточного Web-сервиса (MiddleWare) [4], работа которого заключается в распределении запроса клиента нескольким целевым Web-сервисам с идентичной функциональностью, получении результатов и формировании обобщенного ответа клиенту с повышенными ха-

рактеристиками качества (готовности, достоверности, реактивности и др.).

Распространенным методом повышения достоверности является мажоритирование. Однако недостатком применения этого метода в технологии Web-служб является сложность поддержания жестких схем мажоритирования, таких как “2 из 3”, “3 из 5” и т.д. Одна из ключевых проблем в испытании программного обеспечения – это проверка правильности ответа каждой услуги и определение истинного ответа для запроса. Она может быть решена подсистемой голосования на основе динамически рассчитываемых рейтингов [1, 2].

Кроме того, возникает проблема обработки различных исключений, которые влияют как на функциональность всей системы, так и на достоверность получаемых ответов.

Целью статьи является повышение качества обслуживания (степени доверия к услуге) при работе распределенных приложений на основе совершенствования процедур и способов анализа обработки исключений и мониторинга целевых Web-сервисов.

**1. Анализ факторов, влияющих
на готовность и функциональную
надежность SOA**

В ходе разработки архитектуры [4] при реализации подсистемы рейтингования ответов целевых Web-сервисов, возникла проблема обеспечения го-

товности системы из-за довольно частого возникновения исключений, вызванных различными дефектами. Как будет рассмотрено ниже, возникновение исключений также сильно влияет на величину и динамику рассчитываемых рейтингов. Для построения модели реагирования MiddleWare на исключения было предложено провести исследование исключений, возникающих при динамических клиентских вызовах (Dynamic Interface Invocation, DII) Web-сервисов, на основе которых была предложена реализация данной архитектуры.

Основными группами дефектов при DII являются: дефекты неправильной конфигурации DII Web-сервисов;

дефекты транспортного уровня;

дефекты клиента и целевых Web-сервисов.

Дефекты третьего вида не требуют какой-либо нестандартной обработки, так как не могут быть исправлены посредством MiddleWare.

Проведение же реконфигурации DII при получении исключения первого вида позволит сделать отказы системы устранимыми, а при парировании исключения, возможно уменьшение частоты отказов такого типа (при успешности реконфигурации). Таким образом, повышается среднее наработка на отказ, что повышает готовность системы, которая является важной составляющей гарантоспособности и качества обслуживания [3]. Кроме того, в некоторых случаях парирование исключений при возникновении дефектов второго типа. При парировании исключений такого типа, обеспечивается повышение функциональной надежности транспортного уровня, а следовательно и всей системы.

Таким образом, необходимо провести анализ дефектов для создания модели принятия решения о способе обработки возникающих исключений. Для этого были разработаны приложение и подобран ряд целевых Web-сервисов для сбора статистики по возникающим исключениям. В эту систему были засеяны различные дефекты, по исключениям для которых были собраны статистические данные. Так как

Web-сервисы изначально разрабатывались как программные средства, независимые от языков программирования и платформ, то для обеспечения полноты эксперимента, в системе присутствовали элементы, написанные и опубликованные с использованием различных программных решений.

В табл. 1 приведен список наиболее частых, с нашей точки зрения, дефектов, возникающих при DII Web-сервисов.

Таблица 1
Список исследуемых дефектов

№	Внесенный дефект
1	Обращение к несуществующему сервису (ошибка в URL)
2	Ошибка Target Name Space
3	Ошибка в имени WS
4	Ошибка в Названии порта
5	Ошибка в названии операции
6	Ошибка в типе возвращаемого параметра
7	Ошибка в типе передаваемого параметра
8	Ошибка в имени входного параметра
9	Обращение к заикленному сервису
10	Деление на ноль
11	Ошибка преобразования строки в число
12	Пользовательское исключение
13	Несовпадение числа параметров
14	Несовпадение стиля Web-сервиса ("rpc" и "document")
15	Нет сети
16	Отключен DNS
17	Не запущен сервер
18	Потерян пакет с ответом

В ходе анализа полученных данных было выявлено различное реагирование на дефекты Web-сервисов, развернутых на различных серверах приложений. Для облегчения работы с полученными результатами была составлена классификация, представленная на рис. 1.

В классификации основными являются 2 признака: парируемость дефекта. По этому признаку можно принять решение о необходимых при получении исключения действиях;

этапы возникновения дефекта, на основе которого принимается решение о причине, вызвавшей дефект и его источнике.

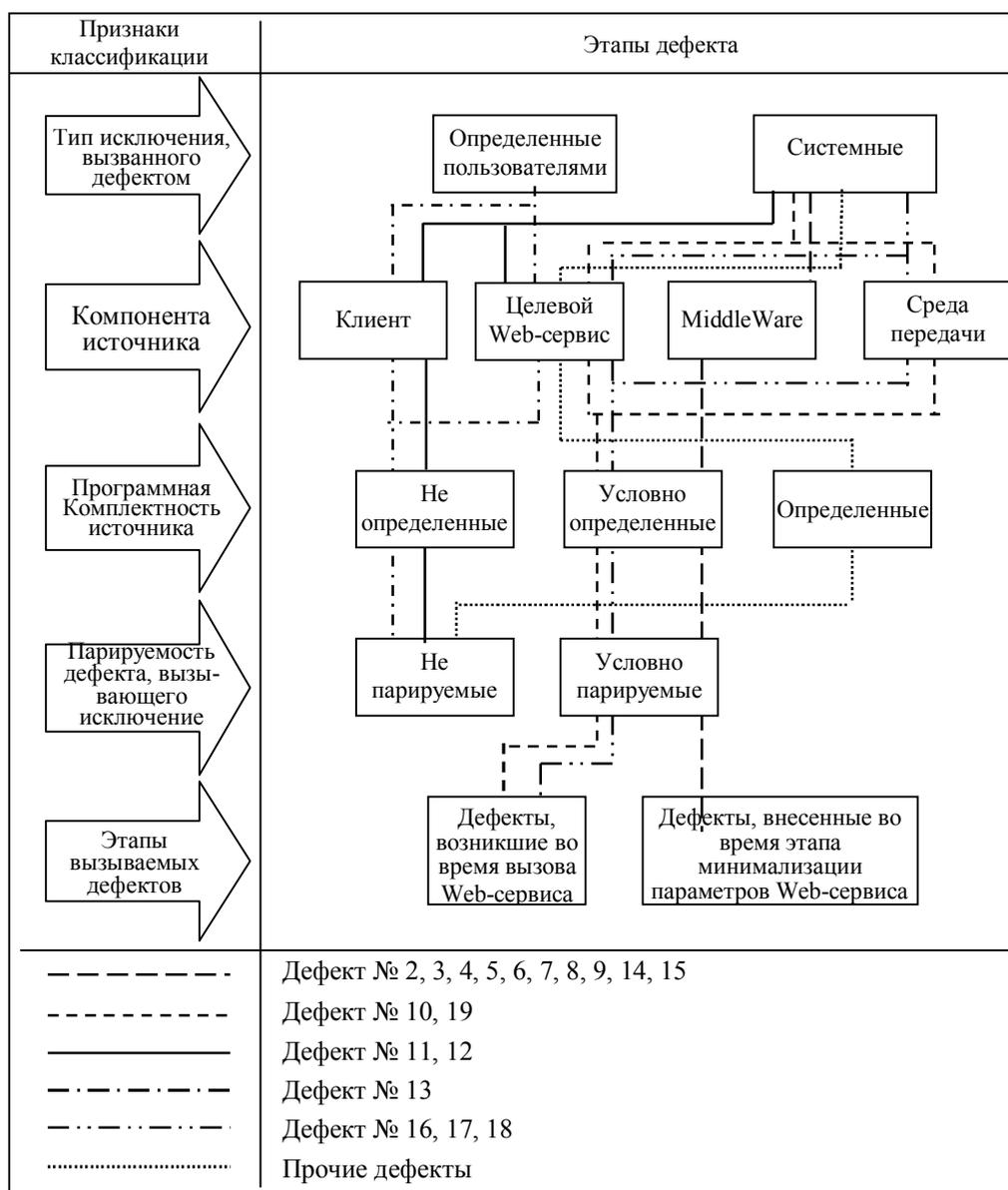


Рис. 1. Классификация дефектов

2. Построение модели поведения MiddleWare при возникновении исключений

С учетом проведенной классификации классификации деление дефектов производится по вероятности их парирования: на дефекты, внесенные при инициализации целевых Web-сервисов (дефекты первого типа), и дефекты, возникшие во время вызова целевых Web-сервиса (дефекты второго типа).

Дефекты первого типа необходимо парировать, для повышения готовности MiddleWare, что реализовано в разработанном приложении.

Парирование дефектов второго типа приводит к улучшению качества обслуживания, так как повы-

шает количество корректных ответов, но может привести к увеличению среднего времени отклика. Исходя из этого необходимость парирования дефектов второго типа целесообразно вынести в настройки для принятия решения об их необходимости клиентом, регистрирующим группу Web-сервисов.

Так как не всегда возможно определить подсистему, дефект которой является источником исключения, то количество попыток парирования целесообразно ограничить.

4. Результаты экспериментальных исследований мониторинга Web-сервисов

Для обоснования целесообразности применения данной архитектуры и используемых методов обра-

ботки исключений был проведен мониторинг четырех Web-сервисов. При этом проводился мониторинг следующих типов:

Без включения подсистем парирования и парирования: с использованием предложенной подсистемы голосования, без ограничения по времени ожидания всех ответов; с использованием предложенных подсистем голосования и парирования исключений, без ограничения по времени ожидания всех ответов; с использованием предложенных подсистем голосования и парирования исключений, с ограничением по времени для нахождения оптимального времени ожидания ответов.

Ниже на рис. 2 – 8 представлены результаты проведения этих мониторингов. В ходе проведения первого этапа мониторинга, был выявлен разброс в значениях ответов целевых Web-сервисов по двум основным причинам, вызванным: формированием исключений, приходящих вместо ответа; различием в ответах, приходящих от различных сервисов при одинаковых входных параметрах.

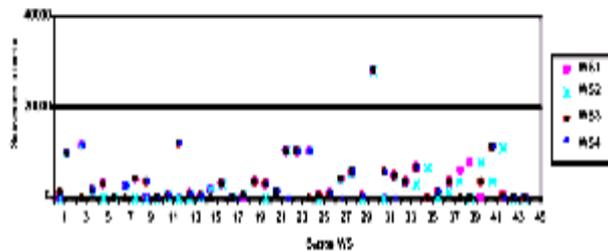


Рис. 2. График разброса ответов целевых WS

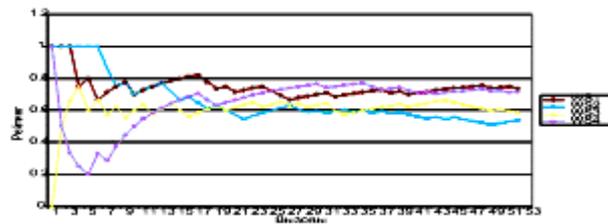


Рис. 3. График динамических рейтингов целевых Web-сервисов 2 этапа мониторинга

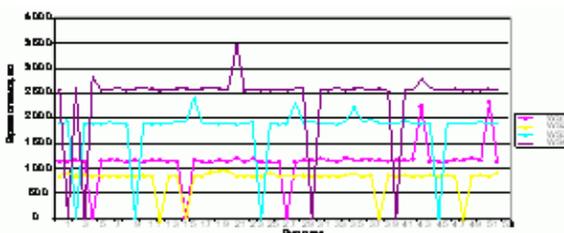


Рис. 4. График времени отклика целевых Web-сервисов (нулевое время – получение исключения вместо ответа) 2 этапа мониторинга

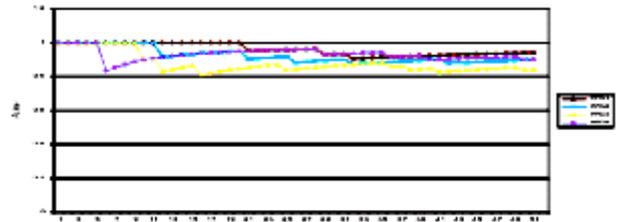


Рис. 5. График динамических рейтингов целевых Web-сервисов 3 этапа мониторинга

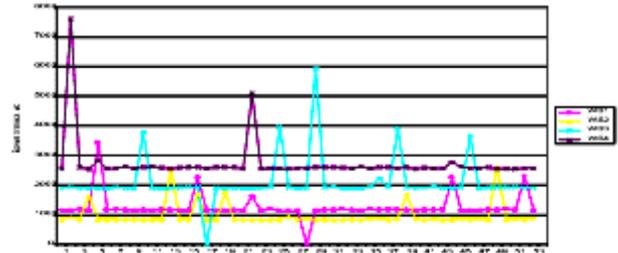


Рис. 6. График времени отклика целевых Web-сервисов (нулевое время – получение исключения вместо ответа) 2 этапа мониторинга

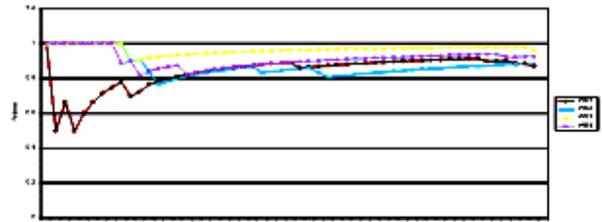


Рис. 7. График динамических рейтингов целевых Web-сервисов 4 этапа мониторинга

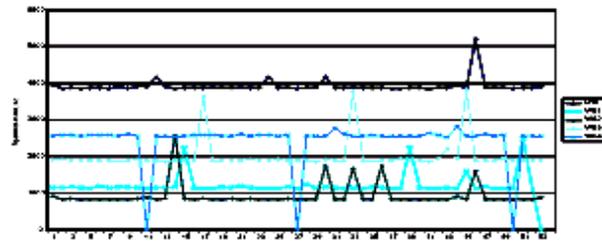


Рис. 8. График времени отклика целевых Web-сервисов (нулевое время – получение исключения вместо ответа) 2 этапа мониторинга

В ходе второго этапа мониторинга была получена статистика по динамическим величинам рейтингов (отношение количества ответов, идентифицированных как правильных, к общему количеству запросов), а также определено среднее число ответов, идентифицированных как правильные, которое составило 2,8 из 4.

Одним из основных факторов, повлиявших на это значение, было большое количество исключений, вызванных дефектами транспортного уровня.

Установлено что, использование подсистемы парирования исключений, которая была использована на третьем этапе мониторинга совместно с подсистемой рейтингов, привело к тому, что: значение динамических рейтингов увеличилось; уменьшилось колебание в их значениях; среднее число ответов, идентифицированных как правильные, возросло в среднем с 2,8 до 3,55; среднее число попыток парирования дефектов составило 1,29 (отношение числа попыток парирования к числу полученных исключений).

Так как в данных экспериментах не учитывалось время работы MiddleWare, а уделялось основное внимание только обеспечению достоверности ответа, на четвертом этапе был проведен ряд мониторингов с различными значениями времени ожидания ответов от целевых Web-сервисов для экспериментального определения оптимального значения величины, при которой обеспечивается наилучшее время отклика без ухудшения остальных параметров. Для обеспечения наибольшей достоверности мониторинг был поведен для различного количества целевых Web-сервисов, от 3 до 9.

В результате было выяснено, что для достижения наилучшего эффекта от подсистемы парирования, который обеспечивает наивысшие показатели при минимальном времени, необходимо чтобы время ожидания было такое, при котором больше половины целевых Web-сервисов можно парировать хотя бы один раз.

Для четырех Web-сервисов среднее число ответов, идентифицированных как правильные, составило 3,48, среднее число попыток парирования дефектов – 1,14.

Выводы

Обеспечение гарантоспособности Web-сервисов(качества обслуживания, степени доверия к услуге) при работе распределенных приложений достигается посредством совершенствования процедур и способов анализа обработки исключений и мониторинга целевых Web-сервисов с использованием динамического рейтингования.

Для реализации архитектуры, позволяющей осуществить такой подход, проведен анализ возник-

ающих при DII Web-сервисов исключений, на основе которого построена классификация дефектов, являющихся причиной их возникновения.

Разработана модель реагирования на возникающие исключения, на основе которой реализована подсистема парирования исключений.

Проведены экспериментальные исследования целесообразности использования архитектуры MiddleWare с применением подсистем голосования и парирования исключений. Их результаты показали, что использование предложенных процедур повышает достоверность результатов и повышает качество обслуживания. Даны рекомендации по определению рационального времени ожидания.

Целесообразно провести исследования по обеспечению выходного контроля для вертикально и горизонтально композуемых Web-сервисов [4].

Литература

1. Townend P., Groth P., Xu J. A Provenance-Aware Weighted Fault Tolerance Scheme for Service-Based Applications // Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'05). – P. 258-266.
2. Tsai W.T., Chen Y., Paul R., Liao N., Huang H. Cooperative and Group Testing in Verification of Dynamic Composite Web Services // 28th Annual International Computer Software and Applications Conference. – Workshops and Fast Abstracts (COMPSAC'04). – P. 170-173.
3. Алмейда В. Производительность web-служб. Анализ, оценка и планирование. – С.-Пб: ООО «ДиаСофт», 2003. – 480 с.
4. Харченко В.С, Горбенко А. В., Боярчук А.В., Мамутов С.С., Михайличенко А.И., Инструментальная платформа для создания гарантоспособных композитных (интегрированных) web-сервисов // Материалы НПК «Информационные технологии – в науку и образование». – Х.: ХТУРЭ, 2005. – С 92-93.

Поступила в редакцию 3.03.2006

Рецензент: д-р техн. наук, проф. В.М. Илюшко, Национальный аэрокосмический университет им. Н.Е. Жуковского "ХАИ", Харьков.