УДК 004.052

А.А. ФУРМАНОВ, И.Н. ЛАХИЖА, В.С. ХАРЧЕНКО

Национальный аэрокосмический университет им. Н.Е.Жуковского, «ХАИ», Украина

МОДЕЛИРОВАНИЕ ГАРАНТОСПОСОБНЫХ СЕРВИС-ОРИЕНТИРОВАННЫХ АРХИТЕКТУР ПРИ АТАКАХ С ИСПОЛЬЗОВАНИЕМ УЯЗВИМОСТЕЙ

Предложена имитационная модель многоверсионной сервис-ориентированной архитектуры, функционирующей при воздействии внешних агрессивных факторов – сетевых атак. Рассмотрены принципы и методика проведения эксперимента с использованием предложенной имитационной модели. Для анализа разрабатываемой имитационной модели предлагается использовать базу данных существующих уязвимостей компонентов сервис-ориентированных систем.

Ключевые слова: *имитационное моделирование, многоверсионная сервис-ориентированная архитектура.*

Введение

Системы с сервис-ориентированной архитектурой (СОА) широко применяются при разработке бизнес-критических приложений, среди которых следует особо отметить Интернет-банкинг, онлайн-магазины, системы резервирования и продажи туристических услуг, системы электронного бизнеса и электронной науки [1]. Однако рост сложности таких систем требует применения новых подходов, обеспечивающих их гарантоспособность. Одним из таких подходов, хорошо зарекомендовавших себя в различных критических областях, является применение многоверсионных технологий [2].

Разработка многоверсионной сервис-ориентированной архитектуры (МСОА) - дорогостоящая задача. Именно поэтому предлагается предварительно построить имитационную модель, при помощи которой можно оценить характеристики гарантоспособности конкретной конфигурации МСОА.

Цель статьи – описание имитационной модели многоверсионной сервис-ориентированной архитектуры и методики её исследования в различных режимах моделирования.

1. Многоверсионная сервис-ориентированная архитектура

СОА работает на определенной компонентной базе (операционные системы, серверы приложений, веб-серверы). Каждый компонент этой цепочки может иметь определенный набор дефектов, внесённых во время разработки.

Основная концепция МСОА (рис.1) заключается в применении принципа диверсности (многоверсионности) [2 – 4] для СОА, который позволяет комплексно решать проблемы обнаружения и парирования отказов, вызванных физическими дефектами аппаратных средств (ФД), дефектами проектирования программных средств (ПД), внешними воздействиями информационного либо иного характера – дефектами взаимодействия (ВД) [2], и уязвимостями [5].

Использование принципа диверсности позволяет решить задачу создания СОА, устойчивой к ФД и ПД, благодаря снижению вероятности отказа по общей причине.

2. Влияние дефектов

Традиционно проявление дефекта при использовании системы (реализации версии) ведёт к ошибке вычислительного или управляющего процесса, т.е. имеет место сбой или отказ и система переходит в неисправное или неработоспособное состояние.

В случае СОА, состоящей из цепочки последовательно соединённых компонент (Web-сервер, Application-сервер, СУБД, ОС), структурная схема надёжности является последовательной, т.е. отказ любого из компонентов системы приводит к отказу всей системы.

2.1. Влияние уязвимостей

Уязвимость – особый вид ПД, незащищённое место в программе, позволяющее злоумышленнику нарушить такие характеристики гарантоспособности системы, как целостность, конфиденциальность, управляемость и доступность.

В работе [5] проанализирована база данных уязвимостей (БДУ) NVD. В этой БДУ каждой обнаруженной уязвимости указывается тип поражения, отражающий характеристику гарантоспособности на которую воздействует уязвимость.

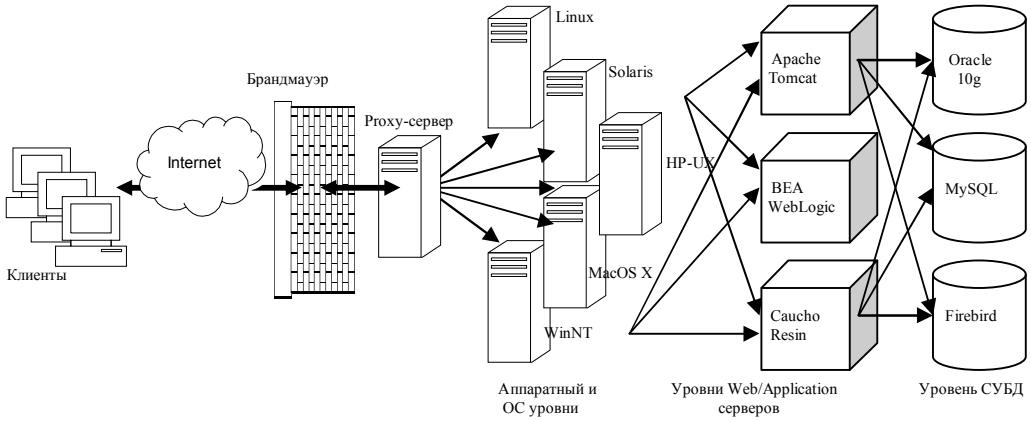

Рис. 1. Структурная схема сервис-ориентированной архитектуры

При подготовке атаки на СОА злоумышленник анализирует программные компоненты, на которых работает атакуемая СОА. Атакующий изучает уязвимости этих компонент и выбирает одну из них, с использованием которой будет производиться злоумышленные действия.

Использование в моделировании реальных данных БДУ в качестве входных данных модели позволяет повысить её достоверность.

2.2. Влияние физических дефектов

Большинство сервис-ориентированных приложений используют аппаратную платформу Intel x86. Её широкое применение и интенсивный подход развития привели к снижению в ней количества ФД.

С другой стороны, постоянно растущая сложность ПО приводит к значительной разнице в количественном отношении ПД к ФД. Поэтому исследовано только влияние ПД, ВД и уязвимостей.

3. Система моделирования

Для построения имитационной модели предлагается использовать технологию Java. Это связано со следующими факторами:

1. большинство существующих реализаций сервис-ориентированной архитектуры представляют собой web-сервисы, построенные с применением технологии Java;

2. технология поддерживает объектно-ориентированную методологию разработки, что придаёт модели гибкость и позволяет её легко расширять, учитывая вновь появляющиеся факторы исследования;

3. программный код имитационной модели может быть легко интегрирован с реальными компонентами СОА, позволяя повысить достоверность исследования;

4. язык Java является широко распространённым языком программирования, что позволит воспользоваться результатами моделирования широкой аудитории исследователей.

Предлагается выделить следующие сущности, используемые в моделировании:

1. задача – основная вычислительная задача, которую может выполнять вычислитель задач

2. уязвимость – программный дефект, позволяющий злоумышленнику произвести атаку

3. атака – задача, приводящая к отказу канала системы

4. генератор заявок – модуль, формирующий поток заявок

5. генератор атак – модуль, формирующий набор атак в потоке заявок, использующий данные о конфигурации системы и производящий последовательность с учётом конфигурации

6. распределитель задач – модуль, распределяющий задачи между различными исполнителями

7. исполнитель задач – модуль, непосредственно выполняющий задачу

8. результат выполнения задачи – структура данных, предназначенная для обработки её мажоритарным элементом

9. мажоритарный элемент – модуль, производящий сравнение полученных результатов и определяющий их корректность

10. анализатор – модуль, фиксирующий результаты работы системы для последующего анализа работы модели

11. конфигуратор – модуль, производящий начальное конфигурирование модели

Упрощённая UML диаграмма классов имитационной модели представлена на рис. 2.

Общая схема имитационной модели (рис.3) отражает размещение модулей и их взаимодействие.

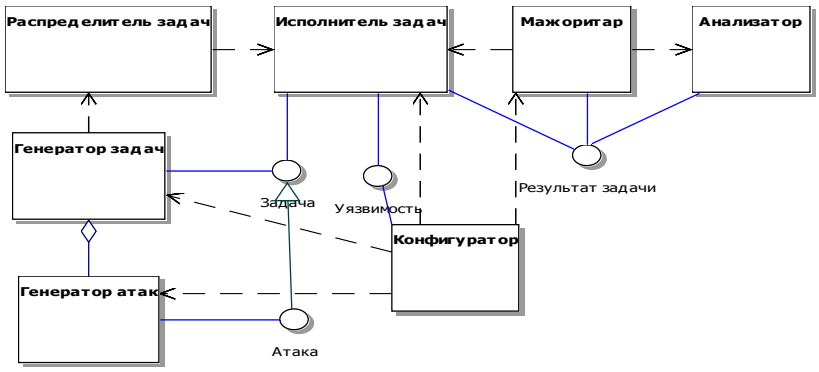

Рис. 2. Имитационная модель (UML диаграмма классов)

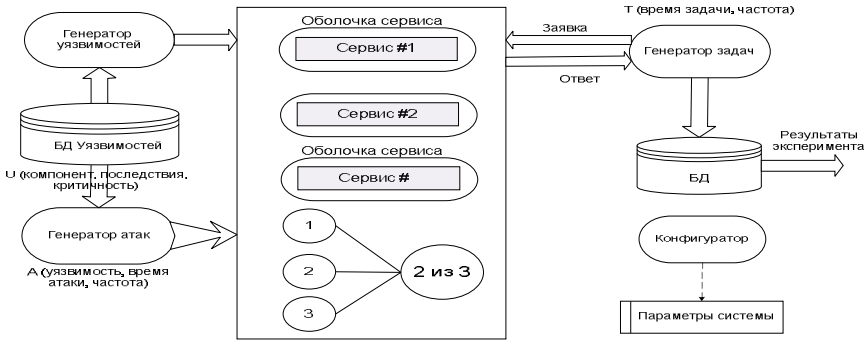

Рис. 3. Общая схема имитационной модели

4. Методика моделирования

Методика моделирования состоит в задании исходных характеристик модели (временных, надёжностных, поведенческих), проведении испытаний модели и последующего анализа полученных результатов её работы.

Алгоритм работы модели включает следующие шаги:

– конфигуратор производит начальную настройку отдельных модулей системы;

– генератор задач выполняет функции клиента СОА, он формирует поток задач для модели, в процессе формирования он обращается к генератору атак и производит засев потока задач атаками;

– генератор атак при помощи БДУ формирует профиль атак в зависимости от конфигурации компонентов МСОА и их уязвимостей;

– генератор задач отдаёт поток задач на распределитель, который, в свою очередь, распределяет задачи между различными исполнителями (каналами);

– после выполнения задачи, исполнители передают результаты на мажоритарный элемент;

– полученные результаты и статистика поступают в анализатор.

Испытание модели целесообразно проводить в двух режимах: *статическом* и *динамическом*.

Статический режим предусматривает исследование работы модели без учета временных характеристик поступления заявок.

При статическом анализе входными данными являются:

– уязвимости, которые содержит каждый компонент СОА;

– атаки, которые проводятся на анализируемые сервисы.

Измерения проводятся для атак с разной кратностью уязвимостей, которая определяет, на какое количество уязвимостей влияет атака. Результатом этих измерений является коэффициент гарантоспособности (1) для каждого из сервисов:

$$k_{гар.} = \frac{К_{атак} - К_{усп.атак}}{К_{атак}}, \qquad (1)$$

где $К_{атак}$ – общее количество атак;

$К_{усп.атак}$ – количество успешных атак.

Важным результатом будет являться разница коэффициентов гарантоспособности различных сервисов и мажоритарного элемента и динамика их изменения.

Кроме того, следует промоделировать ситуацию генерации множественных атак, которые используют уязвимости различных сервисов.

Динамический анализ модели МСОА предполагает учёт временных характеристик потока заявок. При моделировании динамически меняются такие

характеристики: время атаки ($t_{ат.}$), время между атаками ($t_{м.ат.}$), время выполнения задачи ($t_{зад.}$), время между задачами ($t_{м.зад.}$). При каждом запуске системы исследуется влияние изменения одно из этих параметров на гарантоспособность сервиса.

По результатам проведения моделирования необходимо вычислить коэффициенты гарантоспособности СОА, которые представляют собой отношение частот атак к заявкам.

5. Результаты моделирования

Для проведения моделирования были выбраны следующие конфигурации СОА (табл. 1).

Таблица 1

Конфигурации и их уязвимости

Конфигурация	Название продукта	Уровень	Кол-во уязвимостей	Кол-во уязвимостей
1	Apache httpd	WS	14	135
	Tomcat	AS	26	
	Linux	OS	95	
2	IIS	WS	10	158
	WebLogic	AS	35	
	Windows XP	OS	113	
3	Lotus Domino	WS	23	177
	WebSphere	AS	53	
	Solaris	OS	101	

При проведении статического моделирования были выбраны следующие входные данные:

- время атаки – 150мс;
- время генерации атак – 200мс;
- коэффициент уязвимости (под ним понимается число уязвимостей, одновременно используемых для атаки, $K_{уязв}$) – 1 ÷ 4;
- время между снятиями значений – 100мс.

Результаты эксперимента представлены в табл. 2.

Таблица 2

Результаты эксперимента в статическом режиме

		Количество успешных атак			
Ответ сервиса	$K_{уязв.}$	ВС1	ВС2	ВС3	2/3
Сервис недоступен	1	181	212	219	0
	1,25	196	224	183	42
	1,5	227	237	203	89
	2	348	383	356	264
	2,5	356	417	386	320
	3	408	495	440	418
	4	619	726	726	828
Ответ неверный	1	25	16	11	0
	1,25	39	26	18	1
	1,5	60	18	15	3
	2	54	29	19	4
	2,5	37	36	22	4
	3	61	33	20	5
	4	66	42	32	7

С помощью формулы (1) по полученным результатам рассчитаны коэффициенты гарантоспособности и их разница, которые представлены в табл. 3. Разница коэффициентов гарантоспособности позволяет выяснить, когда мажоритирование приносит положительный эффект и когда оно становится неэффективным. График зависимости полученных коэффициентов представлен на рис. 4.

Анализ показывает, что при увеличении коэффициента кратности уязвимости, коэффициенты гарантоспособности плавно уменьшаются. При $K_{уязв}>3$ использование трёхверсионной мажоритированной архитектуры становится нецелесообразным. Это обусловлено тем, что при атаках используется больше уязвимостей, а следовательно существует больше вариантов поражения системы.

Таблица 3

Коэффициенты гарантоспособности сервисов в статическом режиме

$K_{уязв.}$	$K_{гарант.}$				ΔK			
	ВС1	ВС2	ВС3	2/3	$K_{гар.4}$-$K_{гар.1}$	$K_{гар.4}$-$K_{гар.2}$	$K_{гар.4}$-$K_{гар.3}$	$K_{гар.4}$-$K_{гар.ср.}$
1	0,819	0,788	0,781	1	0,18	0,21	0,22	0,20
1,25	0,804	0,776	0,817	0,958	0,15	0,18	0,14	0,16
1,5	0,773	0,763	0,797	0,911	0,14	0,15	0,11	0,13
2	0,652	0,617	0,644	0,736	0,08	0,12	0,09	0,10
2,5	0,644	0,583	0,614	0,68	0,04	0,10	0,07	0,07
3	0,592	0,505	0,56	0,582	-0,01	0,08	0,02	0,03
4	0,381	0,274	0,274	0,172	-0,21	-0,10	-0,10	-0,14

Примечания:
$K_{уязв.}$ – коэффициент уязвимости; $K_{гарант.}$ – коэффициент гарантоспособности; ΔK – разница коэффициентов.

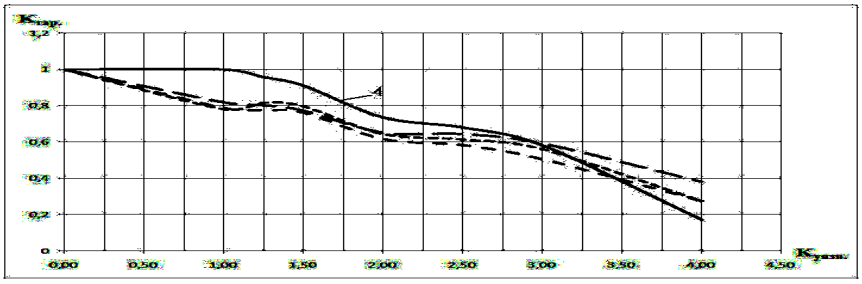

Рис. 4. График зависимости коэффициента гарантоспособности в статическом режиме.

Заключение

Предложенная модель позволяет получить численную оценку характеристик гарантоспособности системы, построенной на базе МСОА для заданной конфигурации её компонентов.

На основе модели следует разработать методику выбора конфигурации многоверсионной сервис-ориентированной системы по заданным критериям.

Достоинством модели является достаточно высокий уровень её достоверности, связанный с использованием реальных данных о дефектах компонентов СОА (БДУ). Данная модель может комплексирована с реальными компонентами сервис-ориентированных систем.

Литература

1. Phifer G. Gartner. Predicts 2008: Web Technologies Continue to Drive Business Innovation – [Электр. ресурс] – / G. Phifer / Режим доступа к ресурсу: http://www.gartner.com/DisplayDocument?id =564307&ref=g_sitelink.

2. Харченко В.С. Многоверсионные системы и обеспечение гарантоспособности. / В.С. Харченко, В.В. Паршин. – Препринт №321. – Х.: ИПМаш, 1989. – 33 с.

3. Gorbenko A. F(I)MEA-Technique of Web-services Analysis and Dependability Ensuring / A. Gorbenko, V. Kharchenko, O. Tarasyuk, A. Furmanov // LNCS 4157, Rigorous Development of Complex Fault-Tolerant Systems / M. Butler et al. (eds.). - Springer, 2006. – P. 153-168.

4. Furmanov A. Intrusion tolerance of Web-systems: IMEA-analysis and multiversion architecture / A. Furmanov, V.S. Kharchenko, A. Gorbenko // Радіоелектронні і комп'ютерні систем. – 2006. – № 7 (19). – С. 23-27.

5. *Furmanov A. The analysis of vulnerability databases for selecting dependable service-oriented architectures / A. Furmanov // Радіоелектронні і комп'ютерні системи. – 2007. – №8(27).– С. 15-19.*

Поступила в редакцию 27.02.2009

Рецензент: д-р техн. наук, проф., зав. каф. компьютерных и информационных технологий и систем А.Л. Ляхов, Полтавский национальный технический университет им. Юрия Кондратюка, Полтава.

МОДЕЛЮВАННЯ ГАРАНТОЗДАТНИХ СЕРВІС-ОРІЄНТОВАНИХ АРХІТЕКТУР ПРИ АТАКАХ З ВИКОРИСТАННЯМ ВРАЗЛИВОСТЕЙ

О.А. Фурманов, І.М. Лахижа, В.С. Харченко

Запропонована імітаційна модель багатоверсійної сервіс-орієнтованої архітектури, що функціонує при дії зовнішніх агресивних чинників – мережевих атак. Розглянуті принципи та методика проведення експерименту з використанням запропонованої імітаційної моделі. Для аналізу імітаційної моделі, що розробляється, пропонується використовувати базу даних існуючих уразливостей компонентів сервіс-орієнтованих систем.

Ключові слова: імітаційне моделювання, багатоверсійна сервіс-оріентована архітектура.

DEPENDABLE SERVIS-ORIENTED ARCHITECTURE MODELING WITH VULNERABILITY USED ATTACKS

A.A. Furmanov, I.N. Lahizga, V.S. Kharchenko

The simulation model of the multiversion service-oriented architecture with influence of external intrusions like network attacks was proposed. Principles and method of simulation using proposed model were considered. Vulnerability database of existed components of service-oriented systems was used for the simulation model analysis.

Keywords: simulation modeling, multiversion service-oriented architecture.

Фурманов Алексей Аркадиевич – ассистент кафедры компьютерных систем и сетей Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: A.Furmanov@mail.ru.

Лахижа Илья Николаевич – студент 5 курса факультета радиотехнических систем летательных аппаратов Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: kenguru@gmail.com.

Харченко Вячеслав Сергеевич – д-р техн. наук, проф., заведующий кафедрой компьютерных систем и сетей Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: V.Kharchenko@khai.edu.