

ОСОБЕННОСТИ ТЕСТИРОВАНИЯ УСТОЙЧИВОСТИ К СБОЯМ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ МЕТОДОМ ГЕНЕРИРОВАНИЯ ОТКАЗОВ

Бородин А.М., Мирвода С.Г., Поршневу С.В.

ФГАОУ ВПО «Уральский федеральный университет им. первого Президента России Б.Н. Ельцина», ул. Мира, 19, Екатеринбург, 620002, Россия, e-mail: sergey@mirvoda.com

В работе рассмотрены проблемы устойчивости корпоративных информационных систем к каскадным отказам. Приведено обоснование проблемы на примере известных отказов в ИС и КИС мирового уровня. Сформулирована основная задача и рассмотрены пути её решения. Приведено сравнение КИС с ИС жизнеобеспечения и безопасности в части самотестирования. Для решения сформулированных проблем предложено авторское программное обеспечение (система СПРУТ), использующаяся для тестирования КИС на устойчивость к каскадным отказам. Приведен опыт использования системы и сформулированы основные требования к подобным системам, обязательные к выполнению при работе в режиме промышленной эксплуатации. Приведены существующие аналоги и показано отличие тестирования КИС в режиме промышленной эксплуатации от тестирования мультимедийных облачных сервисов.

Ключевые слова: тестирование, программный отказ, надёжность информационных систем, внедрение информационных систем.

ON THE RESILIENCE TESTING IN CORPORATE INFORMATION SYSTEMS BY FAILURE GENERATION

Borodin A.M., Mirvoda S.G., Porshnev S.V.

Ural Federal University named after First President of Russia B.N. Yeltsin, Ekaterinburg, Russia (620002, Ekaterinburg, Mira street, 19), e-mail: sergey@mirvoda.com

This paper describes problem of stability in corporate informational systems to cascade failures. Paper presents main problems on a basis of well-known failures in actual world-level corporate informational systems. Paper states main problem and focus on view of its solutions. Comparison of corporate informational systems with systems for life support and security is presented from view of self-testing. As a solution to stated problems, authors propose developed software (system SPRUT) designed to test corporate informational systems on reliability to cascade failures. Paper also describes experience of system's evaluation and formulated requirements to such systems, necessary to execute in process of production exploitation. Existing alternative solutions are listed along with their differences, advantages and disadvantages in testing with corporate informational systems in mode of production exploitation. During this testing focus is made on key differences in testing of cloud services and production corporate information systems. Additionally problem was motivated by providing comprehensive set of examples from real world information systems.

Keywords: software testing, software failure, software deployment, software reliability.

Анализ опыта разработки современных информационных систем (ИС), особенно корпоративных информационных систем (КИС), последнего десятилетия показывает, что де-факто произошел переход к широкому использованию при разработке ИС и КИС программных фреймворков (от английского framework). Началу «эпохи» фреймворков положило появление в 1995 г. и последующее широкое распространение языка Java.

В большинстве случаев фреймворк (каркас, каркасный подход [3]) представляет собой среду исполнения, набор стандартных библиотек и шаблоны использования. Появление фреймворков существенно изменило технологии создания ИС и КИС, в которых раньше стандартные библиотеки использовались лишь для доступа к СУБД, обращения к файловой системе, работе с

кодировками, реализации процедур сериализации/десериализации решения других низкоуровневых задач. Сегодня во всех подсистемах и слоях КИС могут использоваться различные фреймворки.

Анализ опыта их практического использования позволяет сделать вывод о существовании здесь целого ряда проблем. Достаточно часто могут возникать конфликты между различными фреймворками, обусловленные конкуренцией между ними за вычислительные ресурсы ИС или концептуально противоположными подходами, использованными при их разработке [8]. После создания первой версии фреймворка происходит его многократная модернизация, целью которой является наращивание функциональных возможностей. Однако в процессе их модернизации накапливаются устаревшие и/или неверно работающие функции, остающиеся для поддержки обратной совместимости, что чрезвычайно усложняет его использование. Помимо этого разработчики, использующие фреймворки, зачастую не имеют полного понимания их внутреннего устройства.

Перечисленные выше проблемы особенно актуальны для КИС. Действительно, современные КИС с точки зрения системного анализа относятся к сложным системам, поскольку они построены из множества различных подсистем: серверов БД, серверов приложений, кэширующих серверов, балансировщиков, серверов резервного копирования, безопасности и т.д. В хорошо спроектированной системе велика степень взаимной интеграции, например, в технологическую подсистему попадает информация из подсистемы управления персоналом, в подсистему управления персоналом из системы безопасности и т.д. В этих условиях риск выхода из строя одной или нескольких подсистем КИС, за которым следует выход из строя других ранее работоспособных подсистем и так далее вплоть до полного разрушения всей ИС (каскадный выход из работоспособного состояния), оказывается весьма высоким [9].

Отметим, что существует ряд ИС, которые, в отличие от КИС, имеют возможность искусственно сгенерировать отказ и активировать тревогу, например, системы жизнеобеспечения, пожарные и охранные системы. Для проверки степени готовности подобных ИС к реакции «в боевых условиях» обязательно проверяется работа их каскадно-зависимых подсистем. Сегодня в КИС подобные функциональные возможности реализованы, однако, достаточно редко.

При разработке ПО необходимо учитывать, что КИС является сложной динамической системой реального времени, в которой может возникнуть режим каскадного выхода ИС из работоспособного состояния. Без специальных средств обработки нештатных ситуаций ПО окажется неустойчивым к каскадным ошибкам [7] и не сможет гарантировать заложенные в технические требования (ТТ) и соглашение об уровне услуг (SLA) требования по времени восстанов-

ления до работоспособного состояния в случае возникновения предвиденных и непредвиденных ошибок. Для обоснования актуальности проблемы приведем некоторые известные инциденты, произошедшие с ИС и КИС.

1. Сбой 19 августа 2011 г. поисковой системы Яндекс, обусловленный ошибкой программного обеспечения на маршрутизаторе, расположенном в новом дата-центре в Амстердаме.
2. Прекращение работы 26 августа 2008 года большинства банкоматов Сбербанка в Москве из-за отказа процессинговой системы.
3. Сбой социальной сети «ВКонтакте» 27 июля 2014, в результате которого вся информация (фотографии, записи и сообщения, размещенные «ВКонтакте») стала недоступной.
4. Временное зависание на сайте РЖД сервиса по продаже билетов 12 июля 2013 г., связанное с техническими проблемами КИС Транскредитбанка, принимающей платежи за билеты.

Подобные отказы свойственны и крупнейшим Интернет-компаниям. Краткий перечень наиболее известных отказов приведен в Таблице 1.

Таблица 1

Отказы крупнейших интернет компаний

Сервис	Основная причина отказа	Вид отказа	Попытка восстановления	Результат
Amazon EC2	Неверное переконфигурирование серверов	Некорректное секционирование	Множественно некорректное зеркалирование узлов	Отказ множества кластеров
Gmail	Обновление ПО	Несколько серверов отключены	Неправильная маршрутизация	Отключение всех серверов маршрутизации
Gmail	Обслуживание	Отключен один из дата центров	Некорректное кросс зеркалирование дата центров	Почти все дата центры отключились
Google App Engine	Сбой питания	25% серверов не включились	—	Все пользовательские приложения работали с деградированной производительностью
PayPal	Отказ сети	Отключилось ПО верхнего уровня	—	Глобальный отказ в обслуживании
Skype	Перегрузка системы	30 % супер узлов отказали	Образование положительной обратной связи	Отказ всех супер узлов

Из Таблицы 1 видно, что в некоторых случаях, несмотря на то, что во многих программных системах реализован режим самовосстановления после сбоя, данный режим не всегда работает корректно. В ряде случаев имеют место ложные срабатывания, которые могут нанести ИС еще больший вред. Отметим, что каждый из этих инцидентов принес компаниям существенный материальный и имиджевый урон.

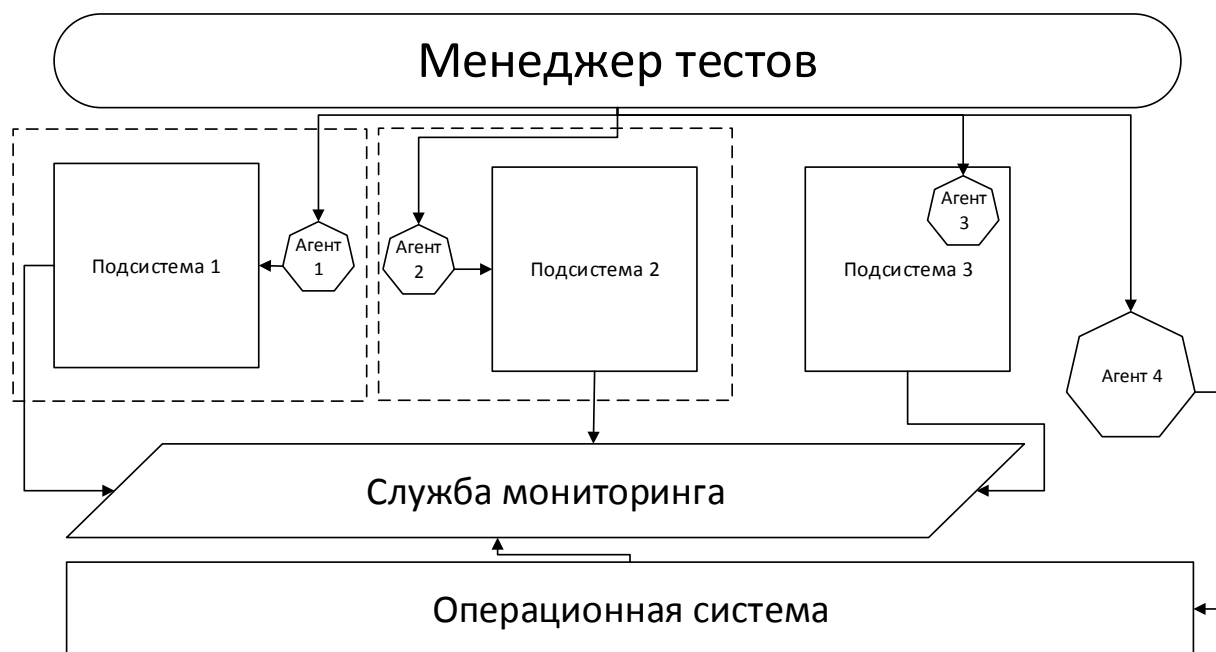
В этой связи разработка методик (сценариев) тестирования КИС, проводимого на этапе ее разработки, является актуальной.

В статье рассматриваются существующие проблемы тестирования и обсуждается система тестирования КИС на устойчивость к каскадным отказам, основанная на использовании сценариев, а также описывается авторский программный инструментарий тестирования, предназначенный для генерирования и каталогизации случайных отказов, включая архитектуру и особенности его использования, приводятся примеры его использования для Автоматизированной Системы Выпуска Metallургической Продукции (АС ВМС). Данная КИС предназначена для решения задач слежения, контроля, моделирования, анализа и выдачи рекомендаций по оптимизации полного цикла выпуска металлургической продукции на основе создания математических моделей технологических, логистических и организационных (бизнес) процессов предприятия.

Разработанная методика и программный инструментарий помогают найти описанные проблемы еще на этапе проектирования и тестовой эксплуатации и, если и не устранить их, то хотя бы предполагать возможность отказа и разработать административные мероприятия по работе в режиме отказа одной или нескольких подсистем.

Архитектура системы проверки устойчивости КИС СПРУТ

Рассмотрим предложенную и разработанную авторами в рамках работ над системой АС ВМП [2, 4, 1] Систему Проверки Устойчивости (СПРУТ), предназначенную для автоматизации тестирования устойчивости КИС к непредвиденным ошибкам, представленную на рисунке.



Архитектура системы СПРУТ

Тестируемая КИС может представлять собой некоторый набор подсистем (на рисунке Подсистема 1, Подсистема N), например, подсистема записи в хранилище данных, подсистема безопасности, пользовательский интерфейс и набор системного ПО (на рисунке – Операционная система). Служба мониторинга собирает информацию о работоспособности тестируемой КИС, в ходе выполнения известных сценариев. Менеджер тестов СПРУТ запускается на всех входящих в КИС подсистемах. Он запускает программного агента, взаимодействующего с ОС по правилам, задаваемым сценарием тестирования, и контролирует его выполнение, просматривает журналы службы мониторинга для определения режима работы Подсистем (сбой, деградация производительности, аварийный режим и т.д.) и ведет собственный журнал результатов выполнения тестов. Сценарии выполняются на том же оборудовании, что и каждая входящая в КИС подсистема. Они могут выполняться как внепроцессно, так и внутрипроцессно (на рисунке Агент 4). Во внутрипроцессном режиме имеются дополнительные возможности влияния на тестируемую подсистему (например, воздействие на рабочую память тестируемого процесса).

Тестом в СПРУТ называется последовательность управляющих воздействий, которая может привести к сбою. Например, для инициирования отказа СУБД из-за недостатка свободного места на диске все свободное дисковое пространство заполняется файлами.

СПРУТ предоставляет возможность выбора управляющих воздействий из каталога, формировать из них тестовые сценарии и запускать их последовательно, параллельно на определенных подсистемах и т.д. Система обладает расширяемой архитектурой, что позволяет тестировать

щикам создавать собственные сценарии вывода КИС из строя, программируя соответствующие управляющие воздействия. Краткий каталог доступных управляющих воздействий представлен в табл. 2.

Таблица 2

Доступные управляющие воздействия на КИС

Подсистема	Управляющее воздействие	Тип запуска
Ввод-вывод	Деградация производительности	внепроцессно
Ввод-вывод	Заполнение свободного места	внепроцессно
Ресурсы центрального процессора	Деградация производительности	внепроцессно
Сеть	Обрыв одного или нескольких сетевых соединений	внепроцессно
Сеть	Задержки отклика	внепроцессно
ОС	Прекращение работы процессов	внепроцессно
ОС	Запуск процессов	внепроцессно
Безопасность	Устананавливает/Снимает разрешения пользователей на некоторые папки	внепроцессно
Процесс	Заполнение памяти процесса	внутрипроцессно
Процесс	Изменение/Разрушение состояния процесса,	внутрипроцессно

Помимо режима работы по сценарию (например, для приёмочных испытаний), СПРУТ может работать в хаотическом режиме, когда применяются случайные управляющие воздействия, и, соответственно, наблюдатель получает возможность изучить поведение КИС в заранее неизвестных условиях. СПРУТ позволяет промоделировать работу КИС в режимах вывода из строя одной или нескольких подсистем (неважно, каким управляющим воздействием), что позволяет проверить устойчивость КИС к каскадному выходу из строя. Например, при отсутствии связи с одним (или всеми) платежными шлюзами тестируемая система должна остаться работоспособной. При этом сообщить пользователю о невозможности выполнить транзакцию и, выбрав доступные варианты оплаты, предложить их пользователю.

Опыт использования СПРУТ показывает, что данная система оказывается наиболее полезной на этапе разработки КИС, так как она, работая в случайном режиме, позволяет обработать большое количество различных сценариев и выявить ситуации, которые трудно, априори, предугадать (например, деградация производительности, падение сетевых соединений, отказы СУБД, разрушение состояния процесса). Кроме того, СПРУТ можно использовать для обучения администраторов КИС: создавая различные отказы КИС, предлагать им принимать решения по устранению возникшей ситуации (подход, аналогичный используемому на учениях по пожар-

ной безопасности), а также проверять срабатывания механизмов отказоустойчивости, резервного копирования и других.

Требования к системам проверки устойчивости КИС

СПРУТ также успешно применялась для тестирования КИС, находящихся в режиме промышленной эксплуатации. Опыт, накопленный здесь авторами, позволяет сформировать список требований, которым должны отвечать подобные системы. Системы проверки устойчивости к сбоям должны обеспечивать:

- безопасность КИС – генерируемые отказы ни при каких условиях не должны повреждать пользовательские данные, приводить к потерям или раскрытию персональных данных или данных для служебного пользования;
- производительность КИС – последствия генерируемых отказов должны устраняться максимально быстро и не оставлять никаких последствий;
- надёжность КИС – отказ должен быть воспроизводимым и сопровождаться исчерпывающим журналом работы.

Заключение

Результаты экспериментальной апробации показывают, что система тестирования отказоустойчивости КИС СПРУТ может применяться и для КИС, находящихся в промышленной эксплуатации. При этом тестирование устойчивости к сбоям данных КИС существенно отличается от аналогичных процедур, проводимых на этапе его разработки. СПРУТ позволяет моделировать те или иные ситуации, выявлять условия выхода КИС из строя, отработать поведение обслуживающего персонала и определить зависимости работоспособности КИС от работоспособности внешних сервисов.

СПРУТ и заложенные в него подходы могут использоваться для приёмосдаточных испытаний в качестве инструмента для обоснования соответствия технических требованиям в части времени выхода на режим в случае известных и неизвестных сбоев.

Отметим, что тестирование устойчивости методом генерирования сбоев было рассмотрено и используется также в [5, 6], однако, указанные методики не могут быть непосредственно применены для тестирования КИС, так как они разработаны и используются для облачных вычислительных сред.

Предложенные подходы и соответствующий программный инструментальный был успешно апробирован при реализации комплексного проекта по созданию высокотехнологичного производства с участием российского высшего учебного заведения по теме: «Разработка автоматизи-

рованной системы слежения, контроля, моделирования, анализа и оптимизации полного цикла выпуска металлургической продукции на основе создания и интеграции математических моделей технологических, логистических и бизнес-процессов предприятия (АС ВМП)».

Работа выполнена в рамках договора № 02.G25.31.0055 (проект 2012-218-03-167) при финансовой поддержке работ Министерством образования и науки Российской Федерации.

Список литературы

1. Аксенов К.А. Метод разработки имитационных моделей реального времени и интеграции с корпоративной системой предприятия / К.А. Аксенов, И.А. Спицина, Е.Г. Сысолетин, А.М. Македонский, О.П. Аксенова // Современные проблемы науки и образования. – 2013. – № 6.
2. Быков Е.А. Аналитический обзор средств и методов для планирования имитационного эксперимента и синтеза мультиагентных процессов преобразования ресурсов / Е.А. Быков, К.А. Аксенов, А.С. Антонова // Современные проблемы науки и образования. – 2014. – № 2. – С. 45-50.
3. Горбунов-Посадов М. М. Расширяемые программы. / Горбунов-Посадов М. М. // — М.: Полиптих, 1999. – С. 336.
4. Aksyonov K.A. Development of real-time simulation models: integration with enterprise information systems / K.A. Aksyonov, E.A. Bykov, O.P. Aksyonova, A.S. Antonova, // Proceedings of ICCGI 2014: The Ninth International Multi-Conference on Computing in the Global Information.
5. Duvall P. Agile DevOps: Unleash the Chaos Monkey. How to deliberately use failure to ensure system resiliency [Электронный ресурс]. – Режим доступа: <http://www.ibm.com/developerworks/library/a-devops4/> (дата обращения: 28.09.2014).
6. Iancutti J. 5 Lessons We've Learned Using AWS [Электронный ресурс]. – Режим доступа: <http://techblog.netflix.com/2010/12/5-lessons-weve-learned-using-aws.html> (дата обращения: 28.09.2014).
7. Gunawi H.S. Failure as a Service (FaaS): A Cloud Service for Large-Scale, Online Failure Drills / Haryadi S. Gunawi, Thanh Do // Electrical Engineering and Computer Sciences University of California at Berkeley. – 28 июля 2011.
8. Manolescu D. A. Why Java is Not Suitable for Object-Oriented Frameworks [Электронный ресурс]. — Режим доступа: <http://micro-workflow.com/PDF/JavaIsNotAFrameworkLanguage.pdf> (дата обращения: 28.09.2014).
9. Richard I. Cook How Complex Systems Fail [Электронный ресурс]. – Режим доступа: <http://www.ctlab.org/documents/How Complex Systems Fail.pdf> (дата обращения: 28.09.2014).

Рецензенты:

Кубланов В.С., д.т.н., доцент, ФГАОУ ВПО «Уральский федеральный университет им. первого Президента России Б.Н. Ельцина», г. Екатеринбург.

Суханов В.И., д.т.н., доцент, ФГАОУ ВПО «Уральский федеральный университет им. первого Президента России Б.Н. Ельцина», г. Екатеринбург.