

Тестирование программных средств

Сафронов Сергей,
2009 год

Оглавление

- Распределенные приложения
 - Плюсы и минусы
- Тестирование распределенных приложений:
 - Тестирование клиентской части
 - Тестирование серверной части
 - Тестирование сети
 - Тестирование связи клиент-сервер
- Тестирование Web-приложений
 - Основные проблемы
- Тестирование Java
 - Основные технические проблемы
- Стресс-тестирование

Особенности тестирования распределенных приложений

- Большие затраты времени
- Требуются эксперты
- Отдельное тестирование клиентской и серверной частей

Плюсы распределенных систем

- Высокая модульность
- Высокая возможность повторного использования
- Наличие внутренних проверок на уровне баз данных
- Возможность использования и интегрирования в систему ПО сторонних производителей
- Возможность балансировки нагрузки клиент/сервер

Минусы распределенных систем

- Огромное количество связей
- Неоднородное окружение (огромное количество вариантов)
- Зависимость от ошибок в стороннем ПО
- Проблемы с безопасностью данных
- Три уровня: клиент, сервер, сеть

Тестирование клиентской части

- Использование затычек для имитации внешних транзакций
- Проверка корректности функциональности клиентской части
- Оценка производительности
- Тестирование в стрессовом режиме:
 - Проверка на низкопроизводительных машинах
 - Запуск на ограниченных ресурсах
 - Ситуация гонок

Тестирование серверной части

- Тестирование транзакций по одной за раз
 - Оценка производительности
 - Оценка корректности работы отдельной транзакции
 - Проверка связности данных
- Одновременный запуск транзакций
 - Оценка производительности
 - Эмуляция многопользовательского режима
 - Проверка блокировок данных и тупиков (deadlock)

Тестирование сети

- Проверка корректности работы и данных в случае:
 - Клиент отключился в процессе транзакции
 - Сеть отключилась в процессе транзакции
 - Кратковременный сбой в сети в процессе транзакции
 - Падение сервера в процессе транзакции
 - Эмуляция перегрузки сети
- Оценка влияния качества сети на производительность

Проверка связи клиент-сервер

- Со стороны клиента:
 - Проверка подсистем
 - Отсылаемые сообщения
 - Всплывающие окна, в том числе сообщения об ошибках
 - Взаимосвязь компонент
 - Проверка взаимосвязей и иерархии данных
 - Переключения между окнами, эмуляция «задумчивого» пользователя
- Работа множественных клиентов
 - Оценка производительности
 - Проверка блокировок и тупиков (deadlock)
 - Параллельная работа с данными

Особенности web-приложений

- Короткие сроки жизни версии
- Постоянно меняющиеся технологии
- Большое число пользователей на начальной стадии запуска
- Доступность 24/7/365
- Проблема Буриданова осла: нагрузка страниц деталями/скорость работы

Тестирование Web-приложений

- Большое число запросов может существенно влиять на производительность
- Работа сервера зависит от скорости сети и используемых протоколов
- Работа со ссылками (потерянные/измененные ссылки)
- Оценка времени отклика для различной скорости сети
- Непредсказуемое число пользователей
- Большое число компонент от различных производителей

Удобство использования для web-приложений

Критерии успешности проектирования:

- Время, требуемое на выполнение задачи
- Число страниц для выполнения конкретной задачи
- В каких местах трудности и с чем они связаны?
- Качество интерактивной справки
- Все ли элементы интуитивно доступны или требуется поиск
- Время загрузки отдельных страниц/элементов интерфейса
- Количество кликов для конкретной задачи, время между кликами
- Страницы, с которых пользователь возвращается обратно

Список основных проблем

- Когда мы видим ошибку со стороны клиента, то мы видим симптом ошибки, но не ее саму.
- Ошибки бывают зависимыми от среды и могут не возникать в различных средах.
- Ошибки могут быть в коде или в конфигурации.
- Ошибки могут постоянно находиться на любом из нескольких уровней.
- Рассмотрение 2 классов операционных сред — статической и динамической — требует различных подходов.

Тестирование Java

- Java applets могут врезаться в HTML страницы (динамический контент)
- При загрузке страницы applet выгружается на клиентскую машину, где и запускается
- Стандартные Java классы:
 - GUI
 - Работа с базами данных
 - Работа с сетью
 - Работа с файлами
- Клиентские классы

Технические проблемы тестирования Java applets

- Версия JDK
- Поддержка различных браузеров
- Поддержка различных платформ
- Корректность отображения на различных платформах/браузерах
- Оценка пиковых нагрузок
- Связь с базами данных (JDBC)
- Унификация тестов для различных GUI библиотек
- Поддержка нестандартных HTML тегов
- Работа с cookies

Поддержка браузеров

- Чем больше целевая аудитория, тем больше вариантов браузеров
- Необходимы проверки
 - Различных браузеров
 - На различных платформах
 - При различных конфигурациях дисплея
- Необходимы проверки HTML-checker-ом

Безопасность

«От любопытного защититься просто, гораздо сложнее защититься от дурака»

- Поиск дыр в логике
- Защита от нелегального доступа (*логи для всех попыток доступа в систему, ограниченное число попыток, время жизни сессии*)
- Защита от sniffers (*https, шифрование*)
- Устранение прямого доступа к файлам
- Защита сохраненных данных
- Очистка «удаленных» данных и памяти

Тестирование безопасности

- Доступ к базам данных
 - Права и ограничения для пользователей
 - Фича: ограничения по времени 😊
 - Права на доступы к объектам/таблицам
 - Целостность транзакций
 - Настройка ролей и групп
- Сеть
 - Защита от Man-in-the-middle
- Уровень приложения
 - Логи для входов в систему
 - Безопасность навигации
- Серверы баз данных
 - Физический доступ к серверам
 - Backup
 - Надежность электроснабжения

Основные элементы GUI

- Меню
- Окно
- Диалоговое окно
- Статический текст
- Кнопка управления
- Селективная кнопка
- Флаговая кнопка
- Поле редактирования
- Окно списка
- Комбинированное окно
- Картинка
- Маска редактирования
- Drag and drop
- Grid

Примеры тестов для диалогового окна

- Информативность заголовка
- Информативность названий кнопок
- Есть ли возможность прервать текущую операцию (и должна ли такая возможность быть)
- Насколько понятен и читаем текст
- Можно ли двигать (и должна ли такая возможность быть)
- Является ли окно модальным

Стресс-тестирование

Проверка системы на предмет пиковых нагрузок

Сложность в том, что есть 4 степени свободы:

- Клиент
- Сервер
- Сеть
- База данных

Примеры наиболее частых жизненных стрессовых ситуаций:

- Возобновление работы после сбоя сервера
- Начало рабочего дня для распределенной организации с жестким графиком работы (банки, биржи)

Стресс-тестирование

Основные направления проверки:

- Работа на слабой клиентской машине
- Одновременная работа сервера с большим числом клиентов
- Корректность работы с большими объемами данных
- Блокировки и тупики при одновременной работе с данными
- Обработка большого числа идентичных запросов