Девятая независимая
научно-практическая конференция
«Разработка ПО 2013»

23 - 25 октября, Москва

2013 CEE-SEC(R)
Software Engineering
Conference in Russia

Performance Test Driven Development

Alexey Ragozin

Deutsche Bank

Тестирование и *нагрузочное* тестирование

Функциональное тестирование

- Количество тест-кейсов увеличивается по мере разработки
- Стоимость ошибки уменьшается по мере тестирования

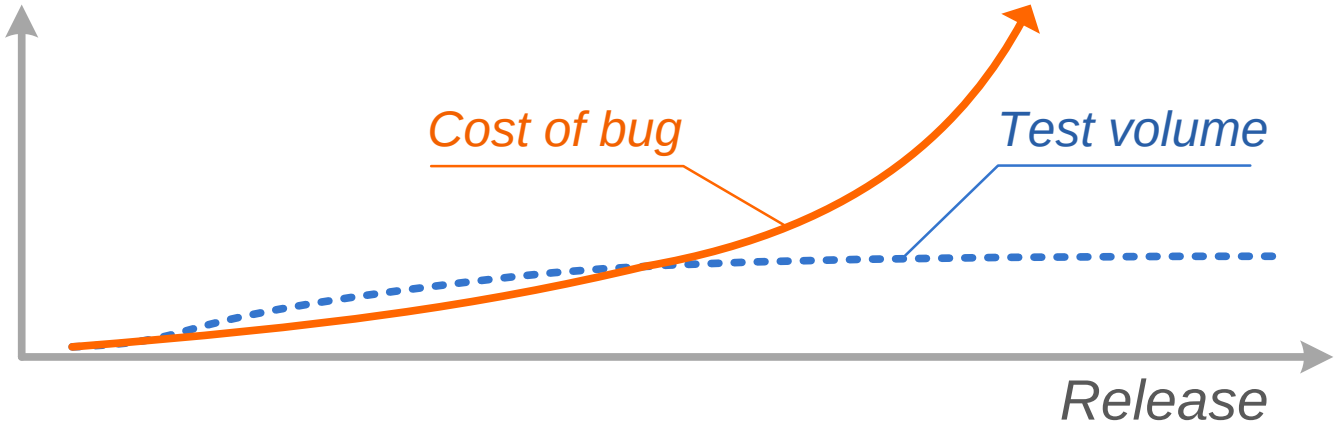
Нагрузочное тестирование

- Количество ***потенциальных*** сценариев тестирования стабилизируется на ранних стадиях разработки
- Стоимость ошибки экспонетциально растёт с ростом функционала

Тестирование и *нагрузочное* тестирование

Функциональное тестирование

Нагрузочное тестирование



Performance Test Driver Development

- Пишем ***неоптимизированный*** код
- Пишем нагрузочные тесты / бенчмарк
- Исправляем проблемы производительности
- Организуем изолированные тесты в профили нагрузки по мере добавления функционала
- Непрерывное тестирование производительности

Оптимизация должна быть обоснована тестом

Никаких спекуляций – только цифры

А на практике?

- Трудоёмкость нагрузочных тестов
  - Сложная логика тестов, распределённые
  - “Ручной труд” в сценариях тестирования
- Отсутствие нагрузочных требованией
  - “Должно работать ***быстро*** и обрабатывать ***много*** данных”
  - Нагрузочный тест план требует отдельного анализа
- Отсутствие адекватной тестовой среды
  - Никто не хочет платить за оборудование дважды
  - Зависимость от внешних компонентов

А на практике?

ВЫ ЖЕ ПРОФФЕСИОНАЛЫ!

ЗАЧЕМ ВАМ НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ?

ПРОСТО СДЕЛАЙТЕ ТАК,

ЧТОБЫ ВСЁ РАБОТАЛО.

И тем не менее

Фундамент для PTTD

- End-to-End автоматизация тестов
- Инкрементальный подход

  бенчмарк → изолированный тест → профиль нагрузки
- Непрерывное нагрузочное тестирование
- Нагрузочное тестирование – ответственность комады разработчиков

Автоматизация

“Классический” подход

- bash + ssh + анализ логов + Excel / R
- Мало пригоден для повторного использования
- Короткий период полураспада тестов
- Использование незнакомого инструментария

“Монокультурный” подход

- Платформа приложения = платформа автоматизации

– Приходится изобретать велосипеды, но

\+ Решается проблема культурного диссонанса

Спектр нагрузочных тестов

- Бенчмарки и распределённые бенчмарки
  - ✓ Проверка гипотез, прототипирование
- Непрерывные нагрузочные тесты
  - ✓ Поддержка тестовой базы в консистентном состоянии
  - ✓ Раннее обнаружение проблем производительности
- Нагрузочные профили
  - ✓ Нагрузочный эквивалент интеграционного тестирования
  - ✓ Проверка соответствия NFR
  - ✓ Профилирование и диагностика проблем

“Правильные” нагрузочные тесты

- Мониторинг, мониторинг, мониторинг
  - Системные и сетевые метрики, тайминги внешних систем и т.д.
- Верификация результатов
  - Эффективность отдачи 503 – не ваш KPI
- Корректность генерации нагрузки
- Качество тестовых данных

Последствия PTTD практики

- Мы стали писать меньше кода
- Тестированием оказались покрыты многие моменты, до которых раньше никогда не доходили руки
- Результаты, полученные на ранних этапах разработки, позволяют более аккуратно планировать закупки оборудования

Открытые проблемы

- Важность нагрузочного тестирования по-прежнему недооценена
- Практически всегда приходится интерполировать результаты из-за ограничений тестовой среды

Ссылки релевантные для Java

Удалённое/распределённое выполнение кода на Java

- *http://code.google.com/p/gridkit/wiki/NanoCloudTutorial*
- *http://blog.ragozin.info/2013/01/remote-code-execution-in-java-made.html*
- *https://github.com/gridkit/gridant*

Статистические расчёты

- *https://sites.google.com/site/piotrwendykier/software/parallelcolt*

Простая библиотека для графиков

- *https://github.com/timmolter/XChart*

Спасибо

Алексей Рагозин

alexey.ragozin@db.com