



Нагрузочное тестирование – автоматизируем полностью

Алексей Рагозин

Постановка задачи



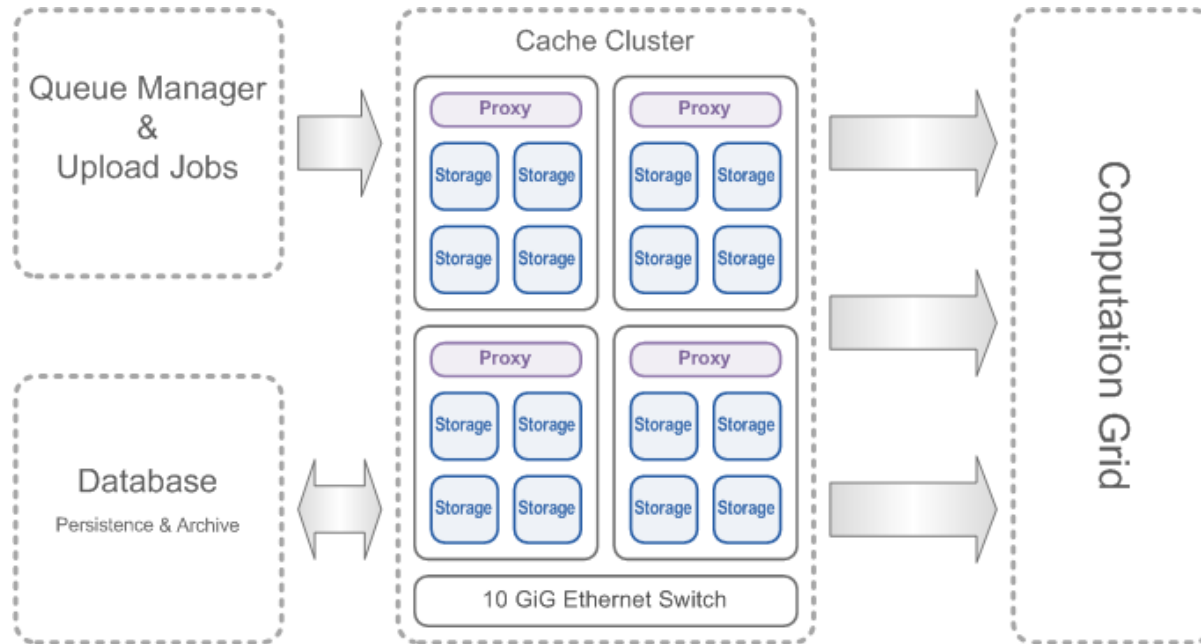
Непрерывное тестирования производительности

- Тестируем последнюю сборку
- Полная автоматизации тестирования
- Тесты запускаются CI сервером

Система для тестирования



Кэш входных данных для вычислительного грида



Нюансы



- ✓ **Тестируемая система распределённая**
Кластер: 4 сервера – 20 процессов
- ✓ **Нестандартный сетевой протокол**
Собственная клиентская библиотека
- ✓ **Для подачи нагрузки требуется ферма**
Мы должны загрузить 4 гигабайтных апплика
Использовался тестовый грид - 50 блейдов
- ✓ **Необходим “прогрев” перед началом теста**
Загрузка тестовых данных
Прогрев JVM

Проблемы распределённых вычислений



Fallacies of distributed computing

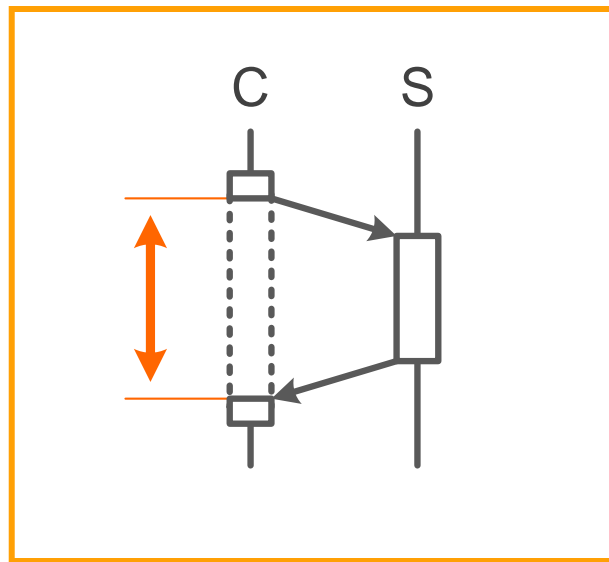
https://en.wikipedia.org/wiki/Fallacies_of_distributed_computing

- The network is reliable
- Latency is zero
- Bandwidth is infinite
- The network is secure
- Topology doesn't change
- There is one administrator
- Transport cost is zero
- The network is homogeneous

Вариации на тему SLA



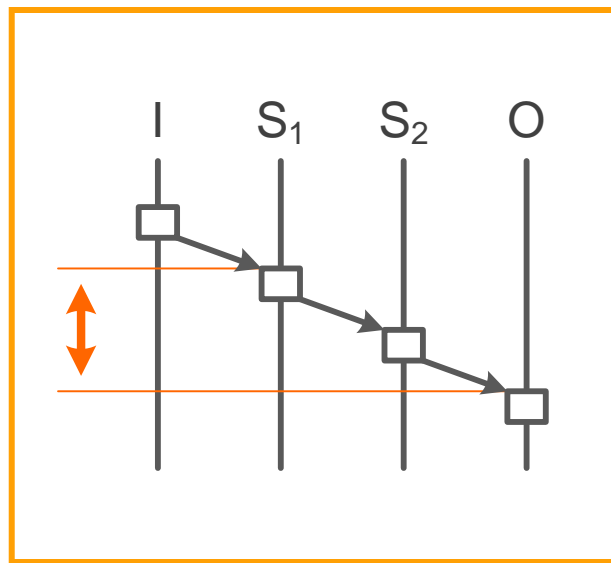
Время отклика



Вариации на тему SLA



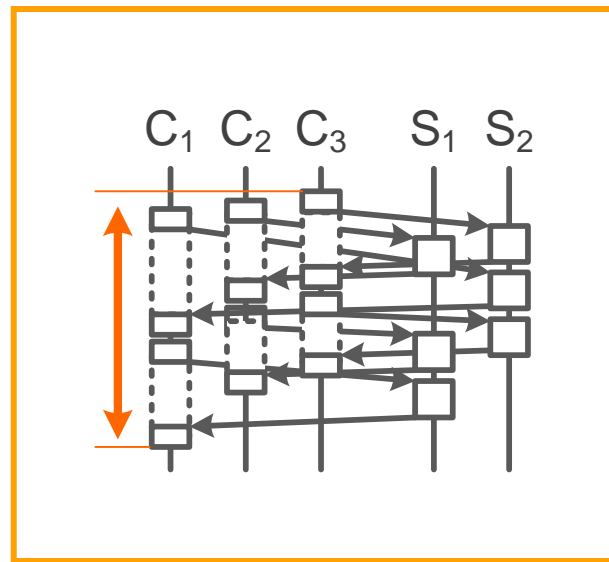
Задержка обработки события



Вариации на тему SLA



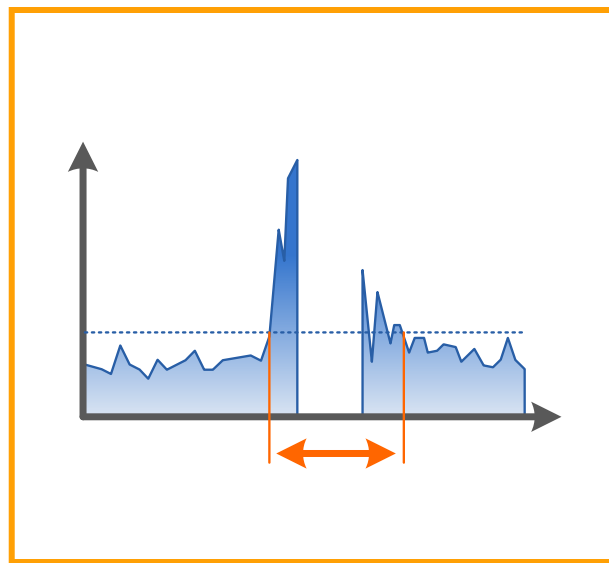
Пропускная способность



Вариации на тему SLA



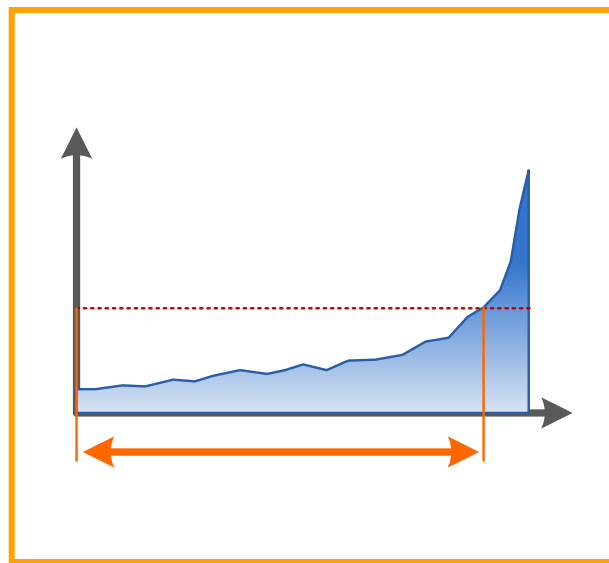
Время восстановления после сбоя



Вариации на тему SLA



Предельно допустимая нагрузка



Этапы тестирования



Подготовка

- ✓ Развёртывание тестируемой системы
- ✓ Развёртывание фермы нагрузки
- ✓ Подготовка тестовых данных / разогрев системы

Прогон

- ✓ Выполнение сценария тестирования
- ✓ Сбор метрик по ходу выполнения

Обработка результатов

- ✓ Агрегация результатов
- ✓ Вычисление ключевых метрик
- ✓ Подготовка детализированного отчёта

Доступные инструменты



- ✓ JMeter
- ✓ Grinder
- ✓ Gatling
- ✓ Яндекс.Танк
- ✓ HP Performance Center

Модульная подход



Задачи

- Разделить и инкапсулировать различные аспекты теста
 - Подготовка тестовых данных / Тестовые агенты
 - Агенты мониторинга / Сбор и обработка данных
- Изоляция ошибок
 - С первого раза всё хорошо не придумаешь
 - Надо иметь возможность переписывания отдельных кусков

Тестирование тестов



Тестовые сценарии надо тестировать и отлаживать!

Уменьшение масштаба теста

- Число узлов / Объём данных
- Время тестирования / Нагрузка

Локальный запуск тестового кода

- Отладка “распределённого” взаимодействия
- Разработка сценария / драйверов

Fail fast



Возможные ошибки

- Ошибки в реализации теста
- Сбои окружения
- Превышение лимитов ожидания

В случае любой ошибки

- Тест должен завершиться немедленно
- Всё процессы должны быть остановлены, ресурсы освобождены
- Исходная причина ошибки должна быть доступна в логе теста

Производительность тестов



Тестовый фреймворк не должен тормозить систему!

Убираем фреймворк с критических путей

- Только управляющие коммуникации
- Передача данных за рамками теста

Управление средой



Задачи

- Развёртывание приложения
- Развёртывание тестовой инфраструктуры
- Zero deployment

Решение

- Nanocloud - <https://github.com/gridkit/nanocloud>
- Удалённое выполнение Java кода по SSH

Nanocloud



Удалённое выполнения Java кода, просто как ...

```
@Test
public void hello_remote_world() {

    Cloud cloud = CloudFactory.createSimpleSshCloud();
    cloud.node("myserver.acme.com").exec(new Callable<Void>() {
        @Override
        public Void call() throws Exception {
            String localhost = InetAddress.getLocalHost().toString();
            System.out.println("Hi! I'm running on " + localhost);
            return null;
        }
    });
}
```

Сбор данных



Проблемы

- Существенный объём данных
- Распределённые источники данных
- Обработка данных не должна влиять на систему

Мониторинг



Зачем дублировать систему мониторинга?

- Единый отчёт по результатам прогона
- Параметры системы могут учитываться при расчёте KPI

Что мониторить?

- Системные метрики используемые при вычислении KPI
- Метрики “здоровья” тестового стенда
System load, IO, Network, etc
- Опционально - данные профилированного приложения

Сбор данных



Решение

- Локальная агрегация данных на узлах
- Передача агрегатов по сети
- Финальная агрегация на управляющем узле

Не заработало!

- Информация теряется
- Сложно описывать правила агрегации заранее
- Работает только с простыми SLA

Сбор данных



Решение

- Локальная буферизация собранных данных
- Файловый буфер / Асинхронная запись
- Передача на управляющий узел по завершении сценария
- Опциональный экспорт в CSV

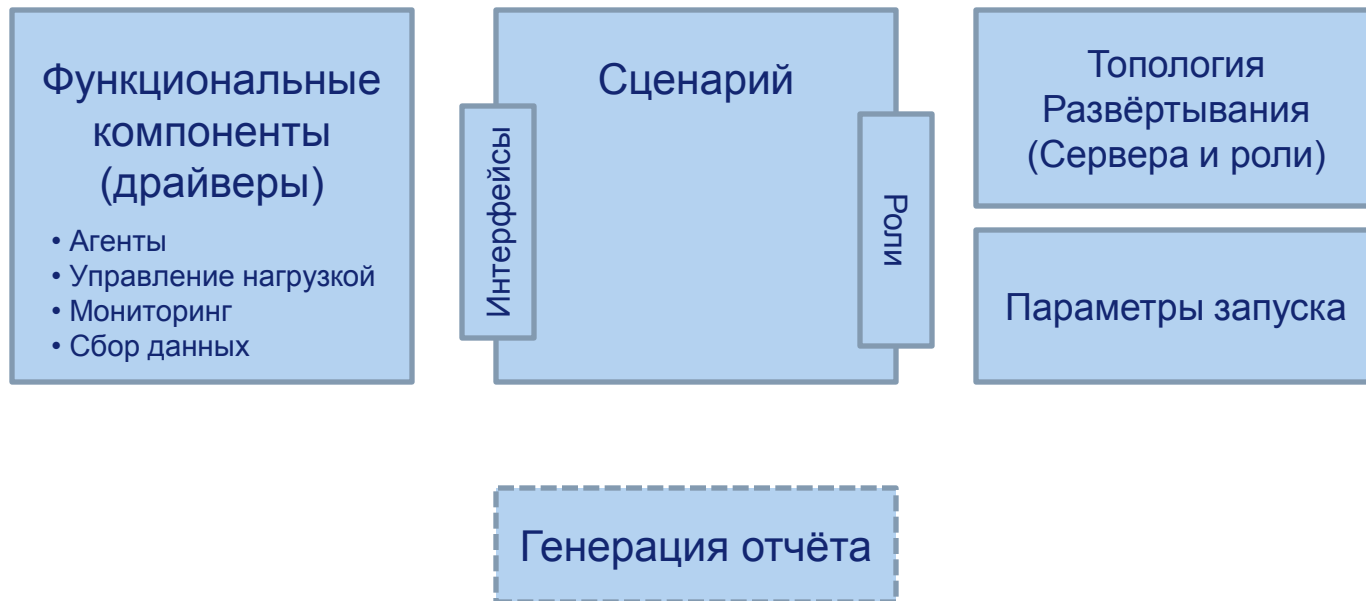
Плюсы

- Детальная информация доступна для обработки

Минусы

- Дополнительная фаза передачи данных
- Использование диска

“Идеальная” архитектура



Mockito



```
@Test
public void test() {

    MyList mock = Mockito.mock(MyList.class,
                               Mockito.RETURNS_DEEP_STUBS);

    Mockito.when(mock.get(0).getValue()).thenReturn("v1");
    Mockito.when(mock.get(1).getValue()).thenReturn("v2");

    Assert.assertEquals(null, mock.get(2).getValue());
    Assert.assertEquals("v2", mock.get(1).getValue());
    Assert.assertEquals("v1", mock.get(0).getValue());

    InOrder io = Mockito.inOrder(mock);
    io.verify(mock).get(2);
    io.verify(mock).get(1);
    io.verify(mock).get(0);
    io.verifyNoMoreInteractions();
}
```


Mockito way



Фаза 1: Декларация намерений

- Моки и вызовы методов
- “Виртуальное” время
- “Абстрактная” распределённость

Фаза 2: Физическое выполнение

- Привязка к топологии
- Зависимости между шагами
- Параллельное / распределённое выполнение

Распределённый сценарий



Fallacies of distributed computing

https://en.wikipedia.org/wiki/Fallacies_of_distributed_computing

The network is reliable – *fail test fast*.

Latency is zero – *small number of round trip*.

Bandwidth is infinite – *low bandwidth control flow*.

The network is secure – *irrelevant for testing*.

Topology doesn't change – *fail test fast and rerun*.

There is one administrator – *zero deployment*.

Transport cost is zero – *not on critical path*.

The network is homogeneous – *location conscious scenario*.

Реальная реализация



Развёртывание & Управление

- SSH + JDK (Linux, Solaris, Windows + Cygwin)
- Java код

Сбор данных

- Локальная буферизация на диск, с последующим копированием на управляющий узел
- csv – формат выходных данных

Обработка и Отчёты

- Нет стандартного решения

Опыт внедрений



Со своим уставом в чужой монастырь не ходят

- Различные цели тестирования
- Различная инфраструктура
- Различная культура

BDD для нагрузочного тестирования



Given Тестовое окружение подготовлено

When Прогон сценария закончен

Then Метрика X лежит в диапазоне [...]

Then Метрика Y лежит в диапазоне [...]

ССЫЛКИ



NanoCloud

- <https://github.com/gridkit/nanocloud/>
- Maven Central: [org.gridkit.lab:nanocloud:0.8.10](https://mvnrepository.com/artifact/org.gridkit.lab/nanocloud/0.8.10)
- <http://blog.ragozin.info/2013/01/remote-code-execution-in-java-made.html>

GridBeans

- <https://github.com/aragozin/gridbeans>

Nimble

- <https://github.com/gridkit/nimble>

Видео доклада о фреймворке

- <http://www.slideshare.net/aragozin/java-57296922>
- <http://www.youtube.com/watch?v=70y83wBGhOU> & <http://www.youtube.com/watch?v=By1reBt7pwg>

Пример бенчмарка

- <https://github.com/gridkit/zk-benchmark-example>

Спасибо



Алексей Рагозин

<http://blog.ragozin.info>

alexey.ragozin@gmail.com



Данный материал не является предложением или предоставлением какой-либо услуги. Данный материал предназначен исключительно для информационных и иллюстративных целей и не предназначен для распространения в рекламных целях. Любой анализ третьих сторон не предполагает какого-либо одобрения или рекомендации. Мнения, выраженные в данном материале, являются актуальными на текущий момент, появляются только в этом материале и могут быть изменены без предварительного уведомления. Эта информация предоставляется с пониманием того, что в отношении материала, предоставленного здесь, вы будете принимать самостоятельное решение в отношении любых действий в связи с настоящим материалом, и это решение является основанным на вашем собственном суждении, и что вы способны понять и оценить последствия этих действий. ООО "Технологический Центр Дойче Банка" не несет никакой ответственности за любые убытки любого рода, относящихся к этому материалу.