

## **Примечание:**

В качестве проекта для создания benchmark для обоснования почему какой-либо фреймворк или СУБД были выбраны в данном проекте, может быть использован любой крупный проект на усмотрение студента – ЛР по курсу Web, курсовая по курсу БД и т.д.

Также допустимо написать отдельное небольшое приложение, которое может сравнивать например различные СУБД на основании запросов из ранее упомянутых проектов, либо сравнивает например два микро-фреймворка для одного языка программирования

## **Лабораторная работа № 3.**

### **Задание:**

Составить набор сценариев (benchmark) для оценки производительности фреймворка для проекта

Примеры часто рассматриваемых сценариев для оценки производительности можно найти например в сравнении разных популярных web-фреймворков:

- <https://github.com/the-benchmarkers/website>
- <https://github.com/TechEmpower/FrameworkBenchmarks>

Так же требуется сравнить хотя бы с одной альтернативой выбранному фреймворку

### **Требования:**

1. Т.к. производительность (не важно – по памяти на сценарий, по времени обработки запроса на сервере, по задержке из-за сериализации объектов, по количеству полных gc-пауз если это jvm, по времени warm-up периода при старте приложения) измеряется статистической величиной, то требуется провести хотя бы 100 испытаний и собрать статистику
2. Т.к. испытаний будет много, желательно добиться равных условий на каждую попытку, то есть следует использовать отдельный докер-образ на каждый прогон, но с одной и той же конфигурацией
3. Следует проверить хотя бы 2-3 параметра: если benchmark для web-фреймворка, то например сериализация объектов + время обработки тяжелых и средних запросов на бэкенде + одновременный логин большого (но допустимого) количества пользователей; если подразумевается непрерывность работы или выбор “самого доступного на данный момент” узла – то время переключения на новый узел в случае загруженности ранее использованного, фактически время ожидания заявки / пользователя / запроса в очереди на ожидание
4. Ожидается такой способ использования benchmark:
  - a. поднимается докер образ с выбранным объектом оценки
  - b. запускается набор тестов для оценки важных параметров
  - c. собирается статистика с такого прогона (на основании лога, либо через условный attach каких-то артефактов к шагам сценария и их анализом,

ключевое – данные должны быть обработаны автоматически либо сгруппированы в .csv, на который можно натравить тот же excel)

- d. выполняем эти шаги 100 раз
- e. собирается окончательная статистика по всем прогонам и формируется отчет
- f. проверяется альтернативный объект тем же способом и формируется отчет

### Пожелание:

Попытаться подумать и сделать вывод об альтернативных фреймворках или даже языке реализации в целом на основании оценки производительности и того, как может измениться производительность в случае масштабирования проекта