

Конфигурация

Тестирование производилось при помощи инструмента k6. k6 позволяет создавать раннеры с заданным количеством запросов в секунду. Заданное количество запросов n в секунду делилось с распределением 90%-9%-1%, соответственно, в секунду исполнялось примерно $0,9n$ contains-запросов, $0,09n$ add-запросов и $0,01$ remove-запросов. В сценариях без дополнительной нагрузки всё время отдавалось запросам одного типа.

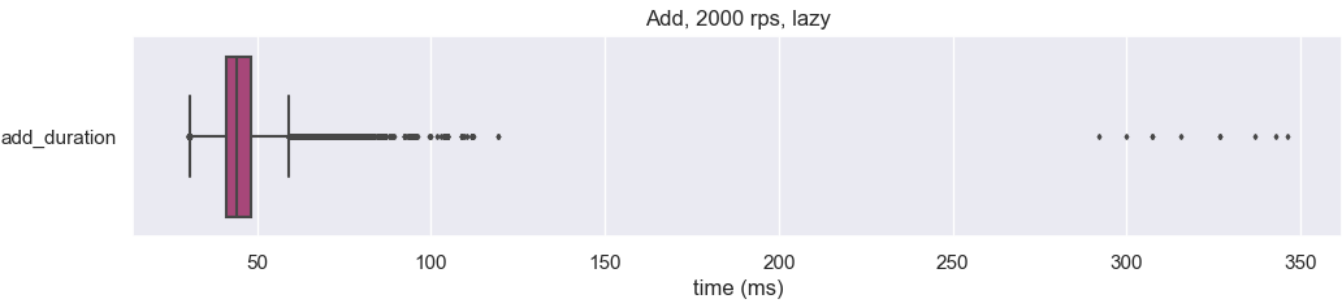
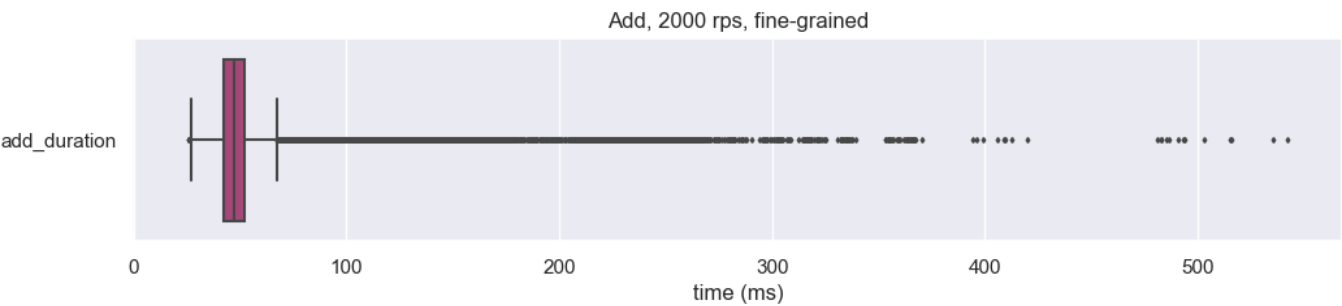
Тестовые данные генерировались случайным образом с заданной верхней границей генерируемых значений. Граница зависела от заданного количества запросов в секунду и длительности теста. Перед запуском каждого теста в систему отдельно добавлялось некоторое количество записей

Конфигурация k6 содержится в файле load-test.js

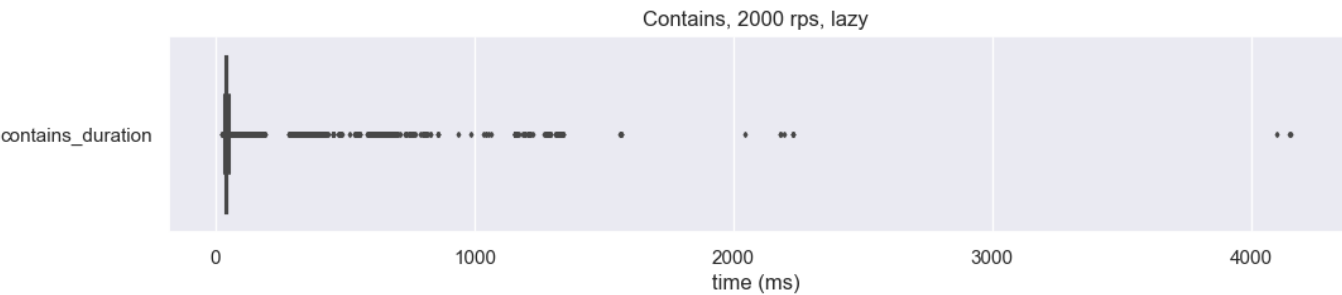
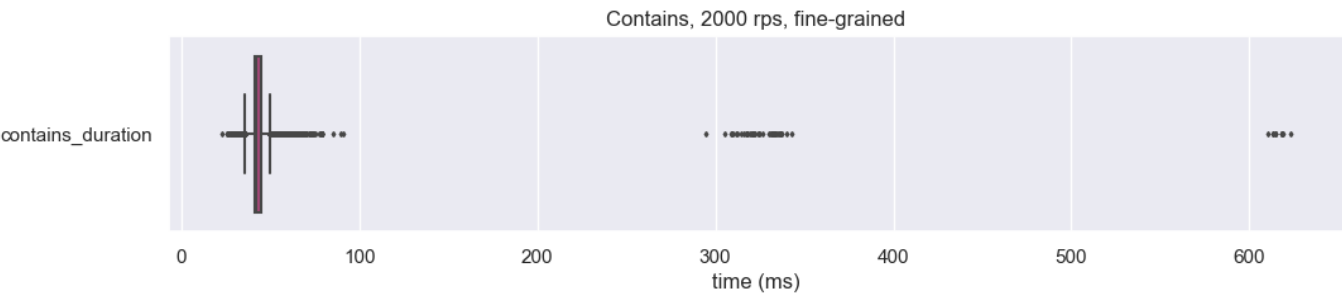
Диаграммы распределения времени выполнения запросов

Длительность тестов, результаты которых приведены далее, составила 30 секунд.

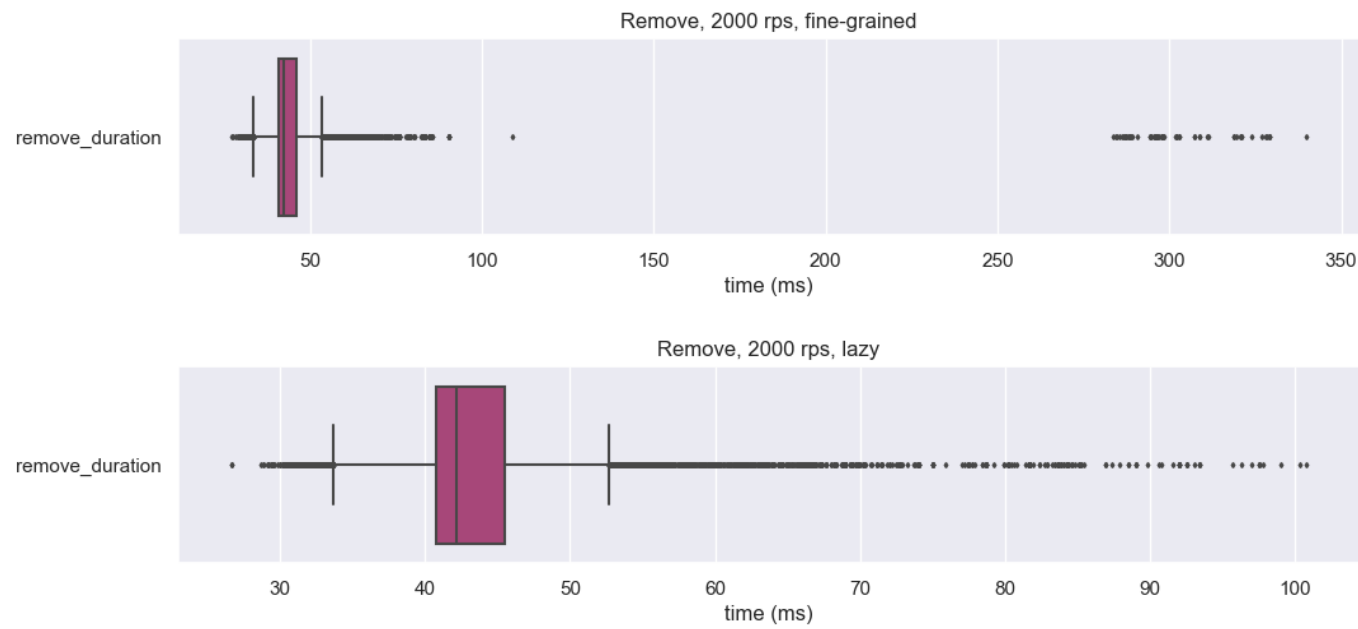
Add в отсутствии другой нагрузки, 2000 запросов в секунду



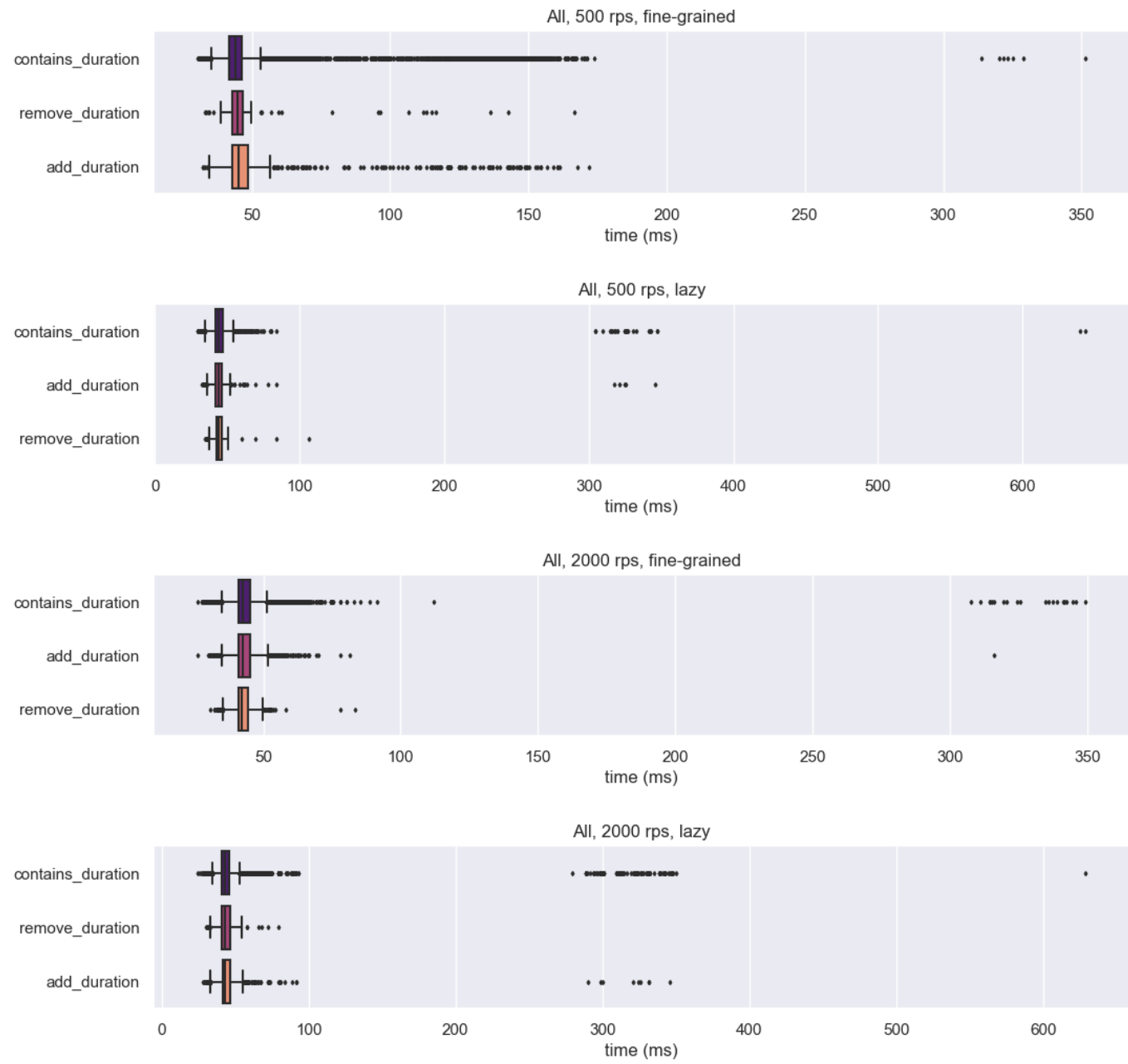
Contains в отсутствии другой нагрузки, 2000 запросов в секунду



Remove в отсутствии другой нагрузки, 2000 запросов в секунду



Все запросы, 500 и 2000 запросов в секунду



Условия отказа от обслуживания

Тестирование на отказ от обслуживания проводилось на сценарии, при котором выполняются и Add, и Remove и Contains. При количестве запросов в секунду, равному 5000, и при длительности теста в 60 секунд начали возникать отказы по таймауту в 10 секунд. Медианное количество записей в системе для теста — примерно 18000 (fine grained) и 16000 (lazy), максимальное — примерно 20000 (fine grained) и 18000 (lazy)