Анализ результатов тестирования Yandex Speller API.

Сопоставление настроек Thread Groups и результатов.

Сравнивая предоставленные настройки Thread Groups с результатами отчета, можно объяснить, почему наблюдаются такие различия в показателях.

**Thread Group 1 (51 пользователя, 1000 итераций)**

Ожидаемое количество запросов: 51 × 1000 = 51,000

Фактическое количество запросов: 51,000 (точное совпадение).

Результаты:

Среднее время: 212 ms

Ошибок: 0.00%

Пропускная способность: 228.7 запросов/сек.

Эта группа показала идеальные результаты без ошибок, что логично при небольшой и постепенной нагрузке (51 пользователь за 1 секунду).

**Thread Group 2 (5000 пользователей, 100 итераций)**

Ожидаемое количество запросов: 5000 × 100 = 500,000.

Фактическое количество запросов: 485,100 (на 14,900 меньше).

Результаты:

Среднее время: 2321 ms.

Ошибок: 24.47% (критически высокий показатель).

Пропускная способность: 1042.5 запросов/сек.

Ключевая проблема: резкое нарастание нагрузки (5000 пользователей за 1 секунду) привело к высокому проценту ошибок. Это указывает на то, что API не успевает обрабатывать такой мгновенный всплеск запросов.

**Thread Group 3 (3000 пользователей, 100 итераций)**

Ожидаемое количество запросов: 3000 × 100 = 300,000

Фактическое количество запросов: 293,801 (на 6,199 меньше).

Результаты:

Среднее время: 2159 ms.

Ошибок: 0.07% (очень низкий показатель).

Пропускная способность: 268.3 запросов/сек.

Несмотря на то, что эта группа имеет больше пользователей, чем Group 1 (3000 против 51), ошибок гораздо меньше, чем в Group 2. Причина: более плавное нарастание нагрузки (3000 пользователей за 10 секунд вместо 1 секунды).

Ключевые выводы

Способность к адаптации под нагрузку:

API Yandex Speller хорошо справляется с постепенно нарастающей нагрузкой (Thread Group 3: 0.07% ошибок).

Не справляется с резким всплеском запросов (Thread Group 2: 24.47% ошибок.)

При плавном увеличении нагрузки (10 секунд на 3000 пользователей) система стабильна.

При мгновенной нагрузке (1 секунда на 5000 пользователей) система деградирует.

Если приложение будет генерировать плавный рост трафика, API будет работать стабильно.

Если трафик будет резко возрастать (например, при запуске новой функции), возможны проблемы с ошибками.