Нагрузочное тестирование

Цель

Целью нагрузочного тестирования является проверка производительности системы и ее способности справляться с повышенной нагрузкой. В ходе тестирования осуществляется имитация большой нагрузки на систему, различных пользовательских сценариев и максимальной нагрузки на ресурсы системы.

Главная цель нагрузочного тестирования состоит в проверке границ работы системы и обнаружении ее слабых мест, которые могут привести к сбоям, выходу из строя или другим непредвиденным последствиям при существенно увеличенной нагрузке на неё.

Подготовка к тестированию

Перед началом тестирование создадим в базе по 5000 пользователей для ролей игрока, организатора и издателя. Для этого создадим сценарий со счетчиком 1 до 5000 тысяч и создадим три запроса в потоке, которые создают пользователей с разными ролями.

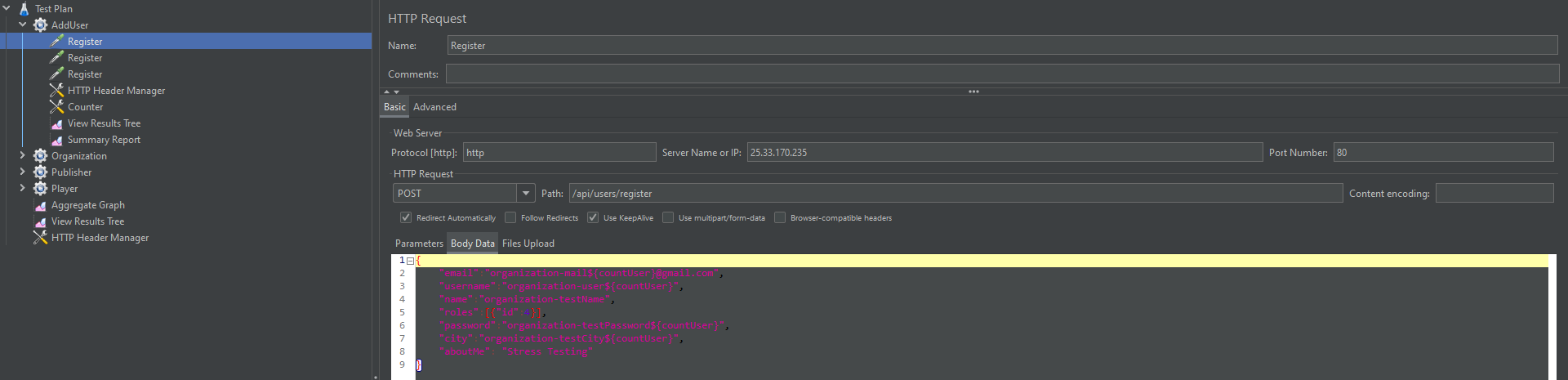


Рисунок 1 Сценарий создания пользователей

Далее тестирование будет проводиться методом ступенчатой нагрузки. Тестирование будет проводиться отдельно для каждой роли по их типовым сценариям в системе. Сценария будут описаны далее.

Организатор

Организатор пользователь, который создает мероприятия. Поэтому сценарий будет достаточно простым, мы авторизуемся под ролью пользователя и создадим мероприятие.

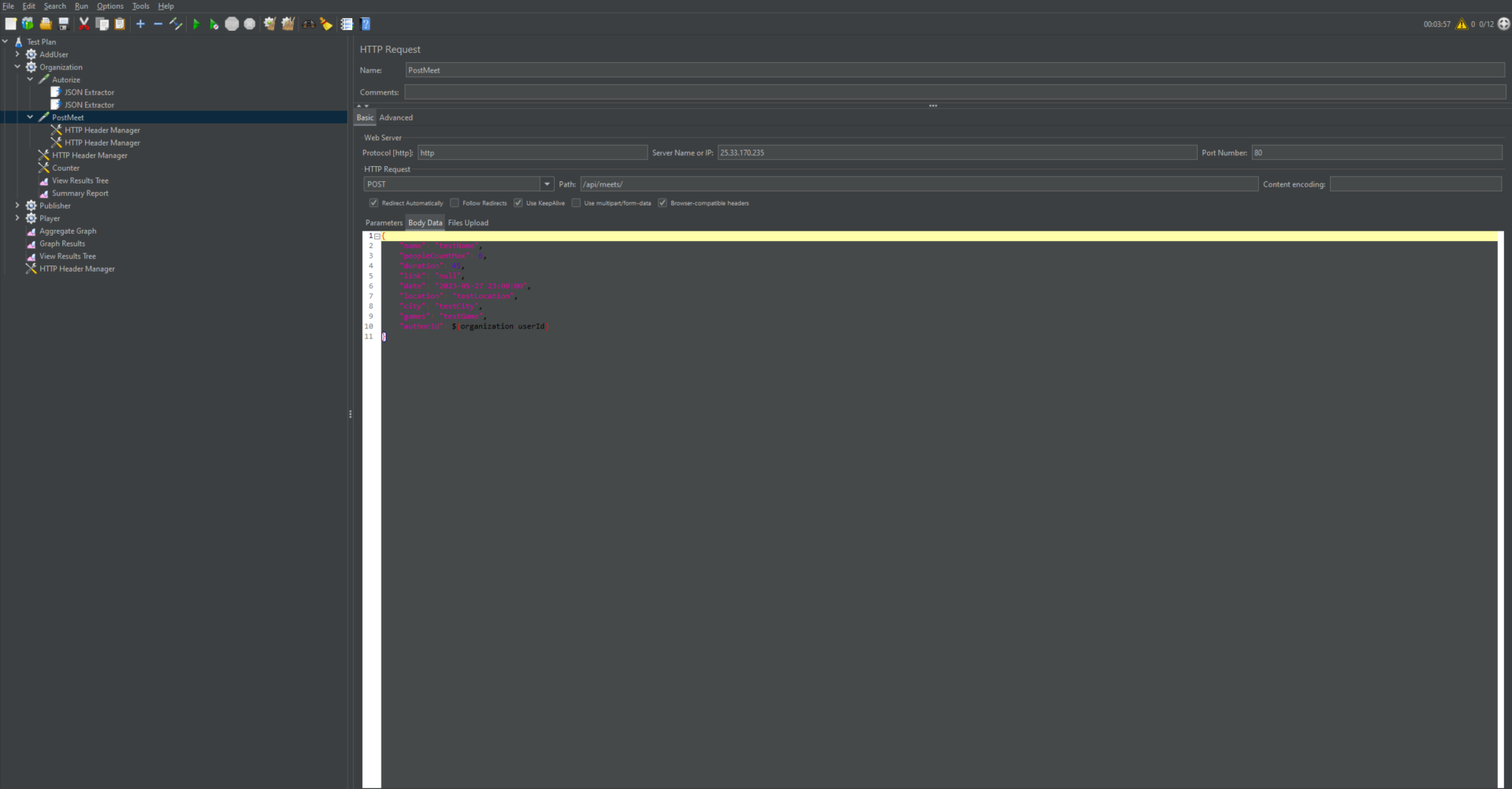


Рисунок 2 Сценарий организатора

Начнем с 10 пользователей. Как мы видим система полностью стабильна поэтому увеличивает количество пользователей в два раза и продолжаем тестирование

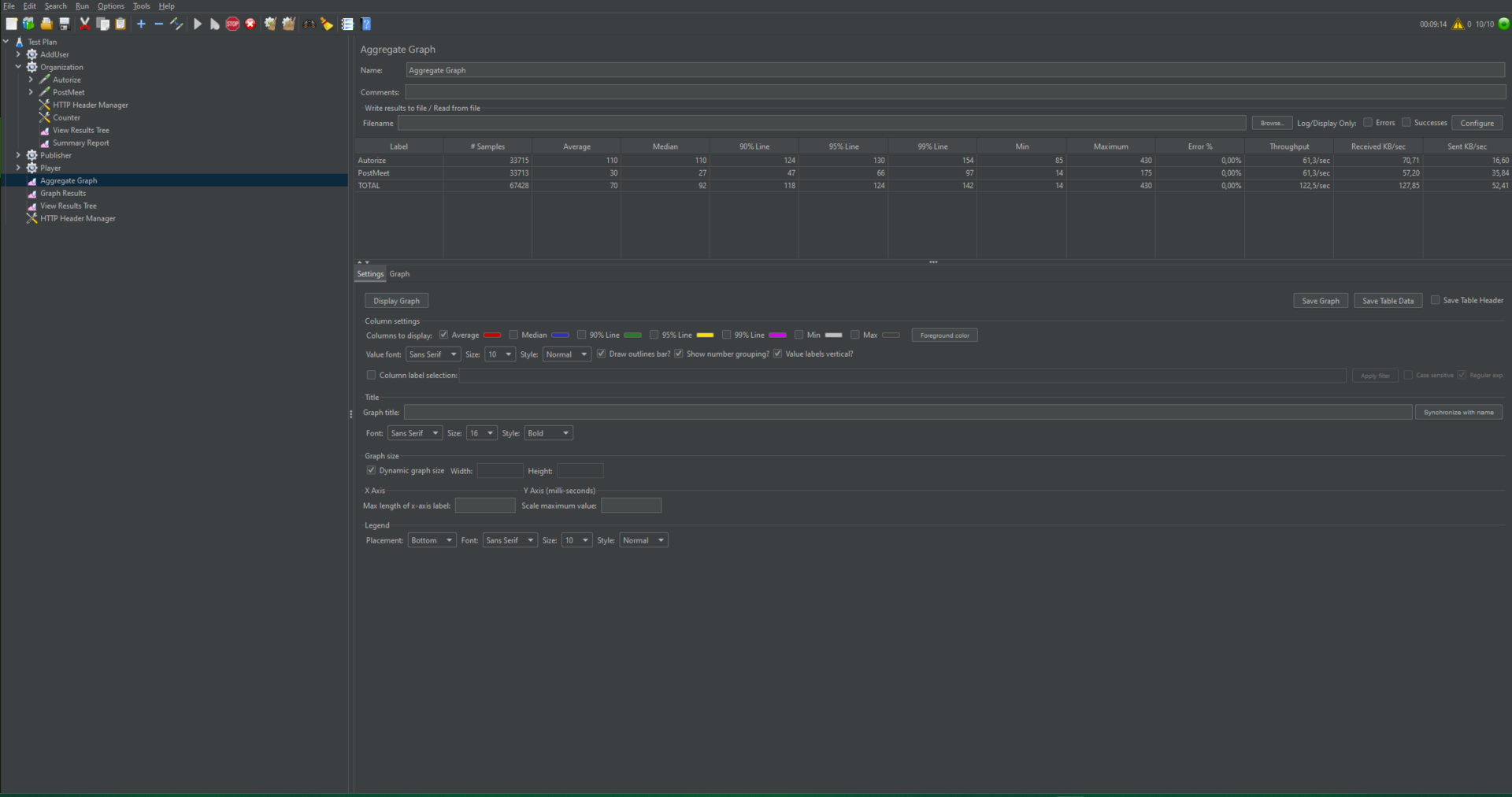


Рисунок 3 Результат при 10 организаторах

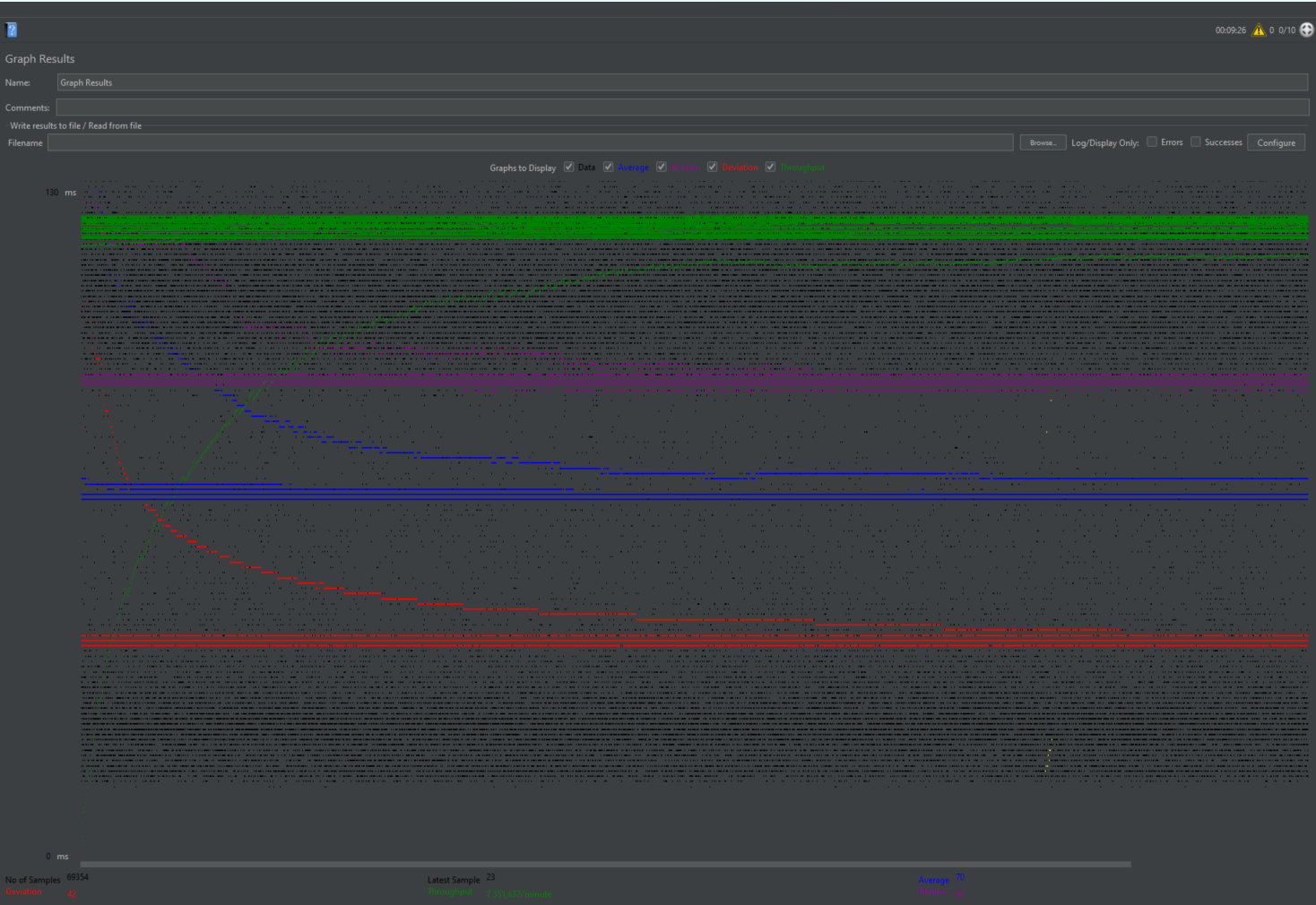


Рисунок 4 Результат при 10 организаторах

При 20 пользователях система стала работать менее стабильно, и 5 процентов всех запросов начала отклонять и выдавать ошибку. Ошибка состоит в том, что JDBC не может выдать подключение из-за того, что они все заняты. Так как на большинство запросов система отвечает, мы продолжаем тестирование и увеличивает количество пользователей в 2 раза.

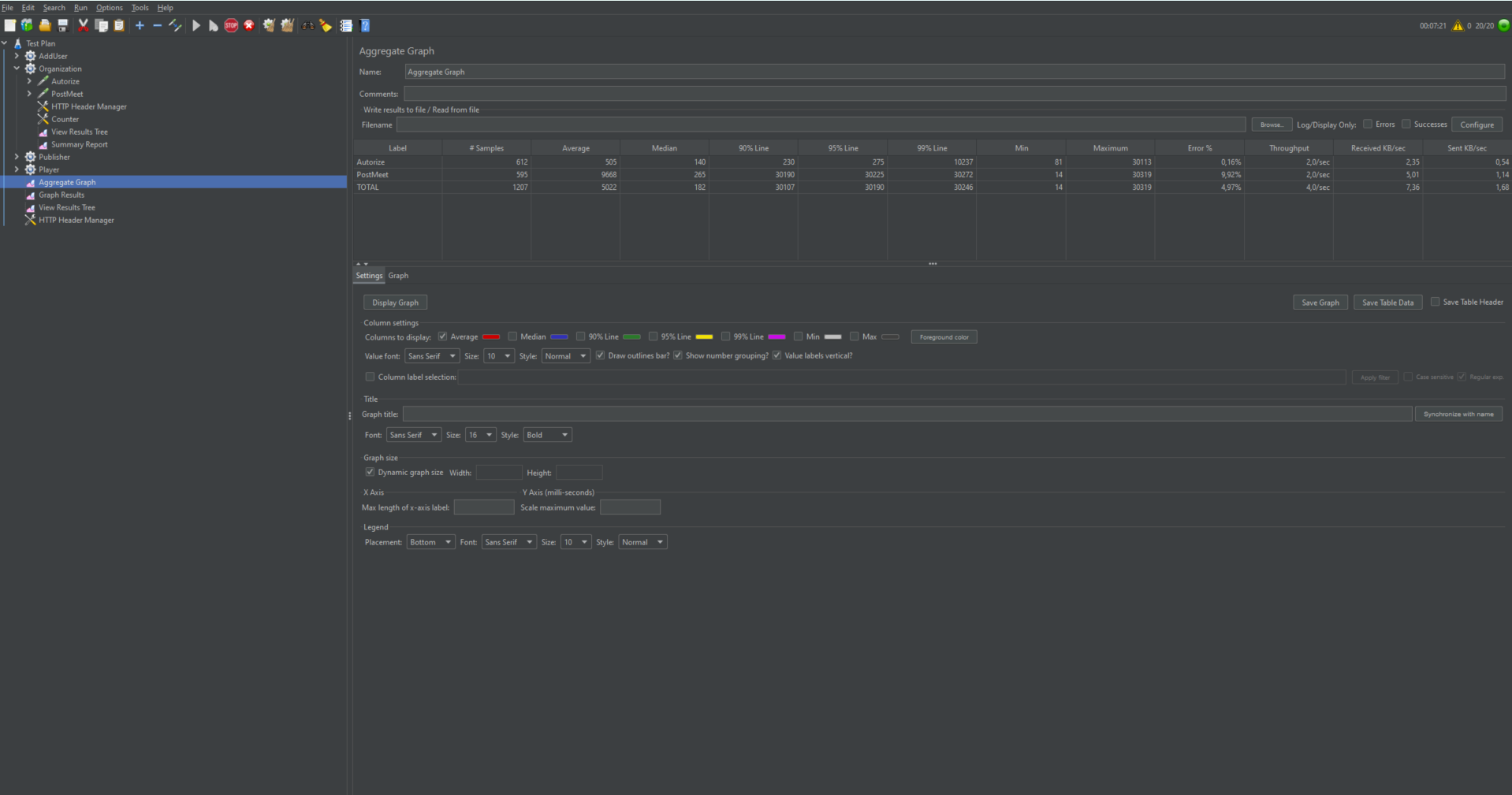


Рисунок 5 Результат при 20 организаторах

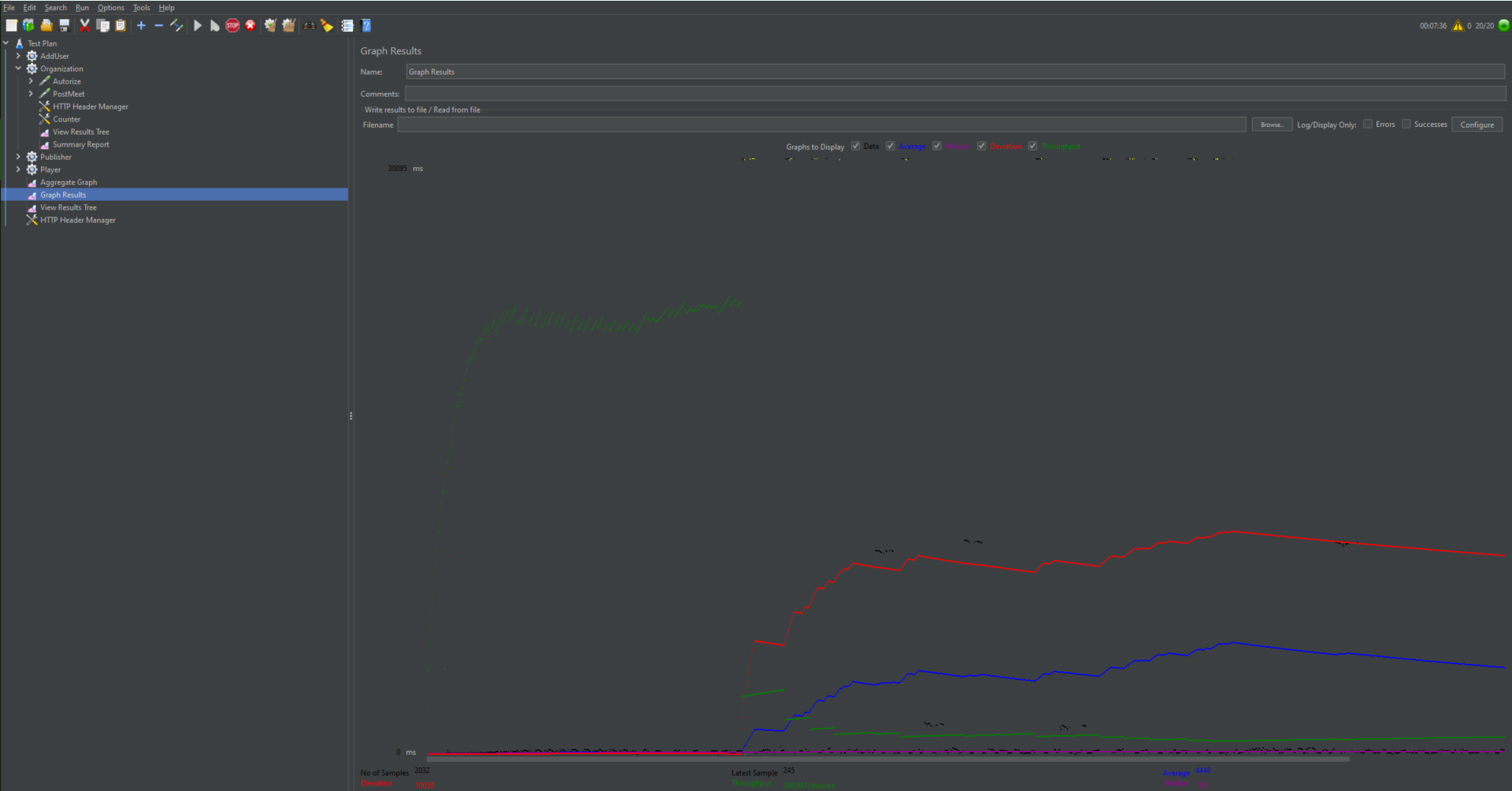


Рисунок 6 Результат при 20 пользователях



Рисунок 7 Ошибка JDBC

На 40 пользователях система стала работать очень нестабильно, и запросы возвращаются с то ей же ошибкой. Поэтому по методу ступенчатого тестирования, возьмем 80 процентов пользователей и протестируем 15 минут.

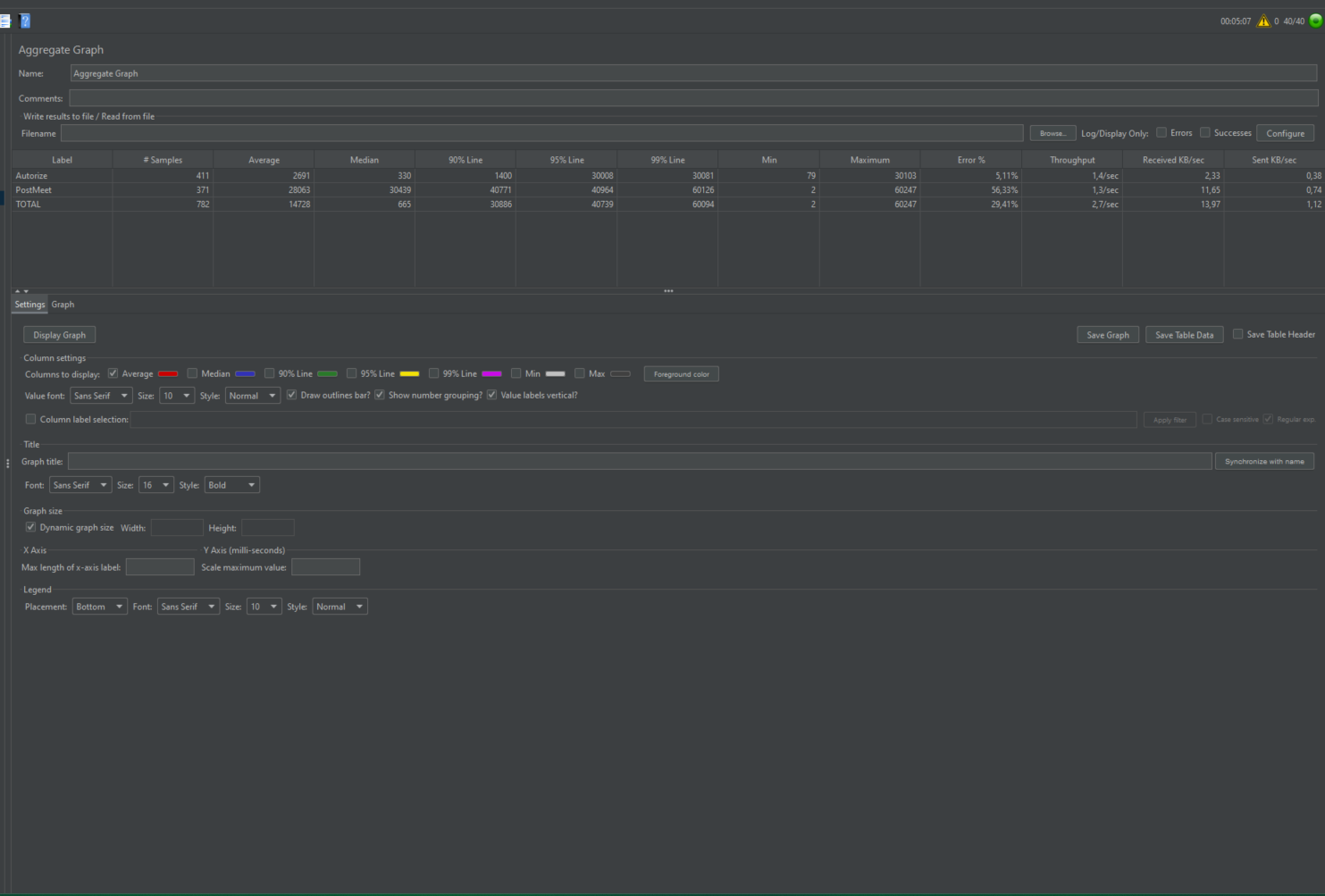


Рисунок 8 Система при 40 пользователях

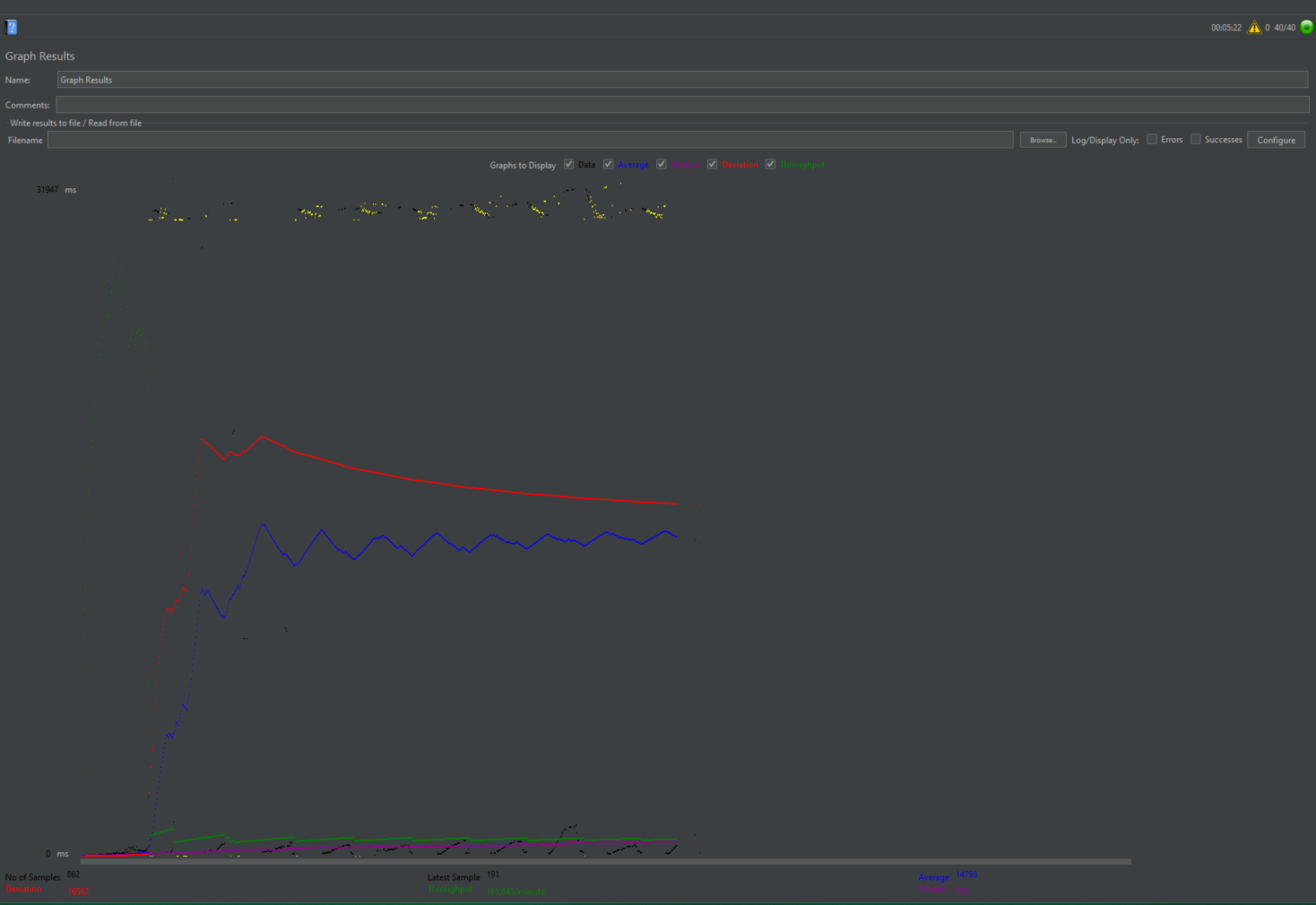


Рисунок 9 Система при 40 пользователях

При 32 пользователях за 15 минут система показала себя более стабильно, чем при 40 пользователях. Она смогла ответить на более 50 процентов запросов.

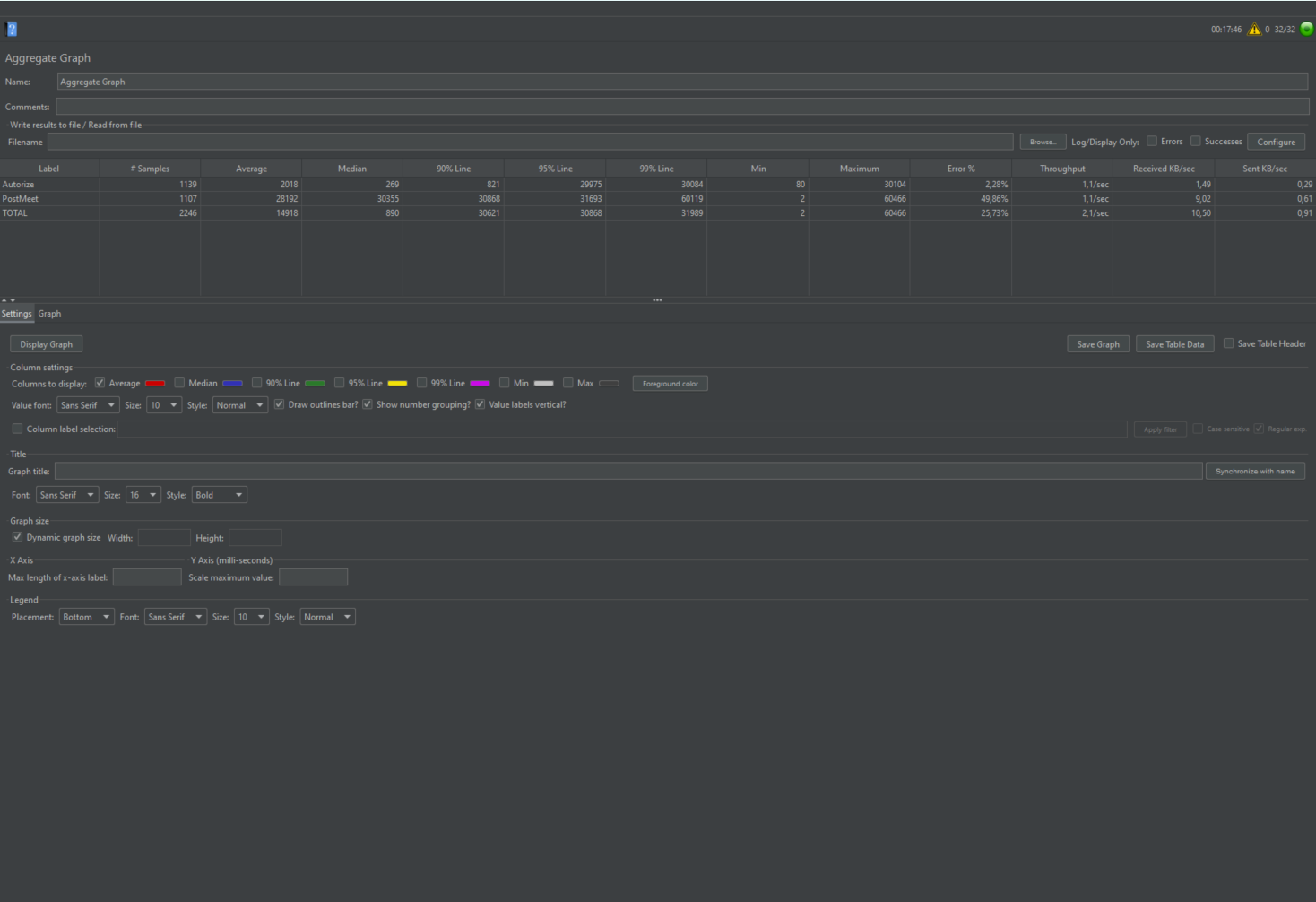


Рисунок 10 Система при 32 организаторах

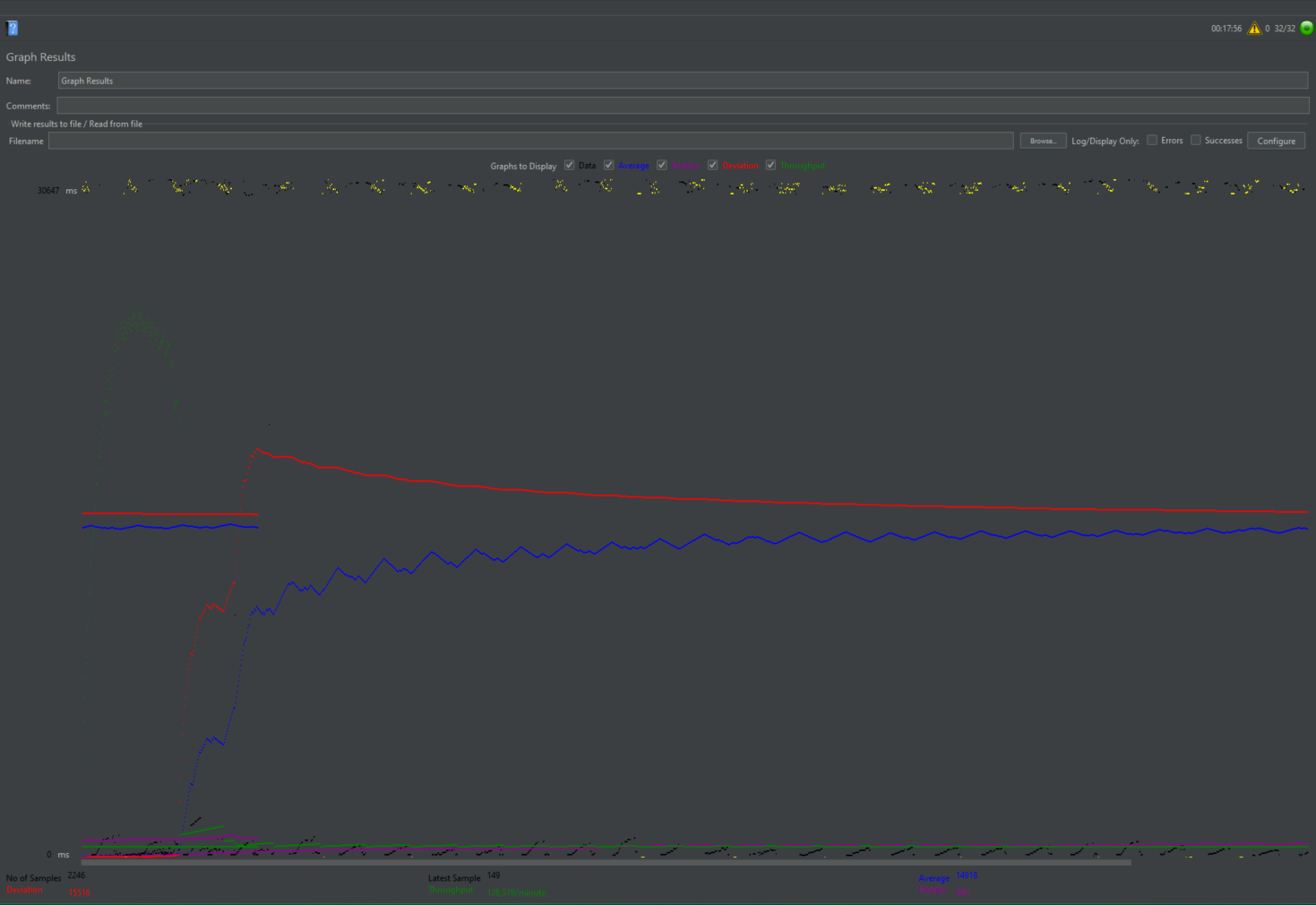


Рисунок 11 Система при 32 организаторах

Издатель

Издатель пользователь, который наполняет систему статьями о настольных играх. Его сценарий состоит из 3 пунктов авторизации, публикации мероприятия и изменения мероприятия. Начнем мы также как и раннее с 10 издателей.

При 10 издателях работает стабильно и не выдает никаких сбоев. Поэтому увеличим количество издателей до 20.

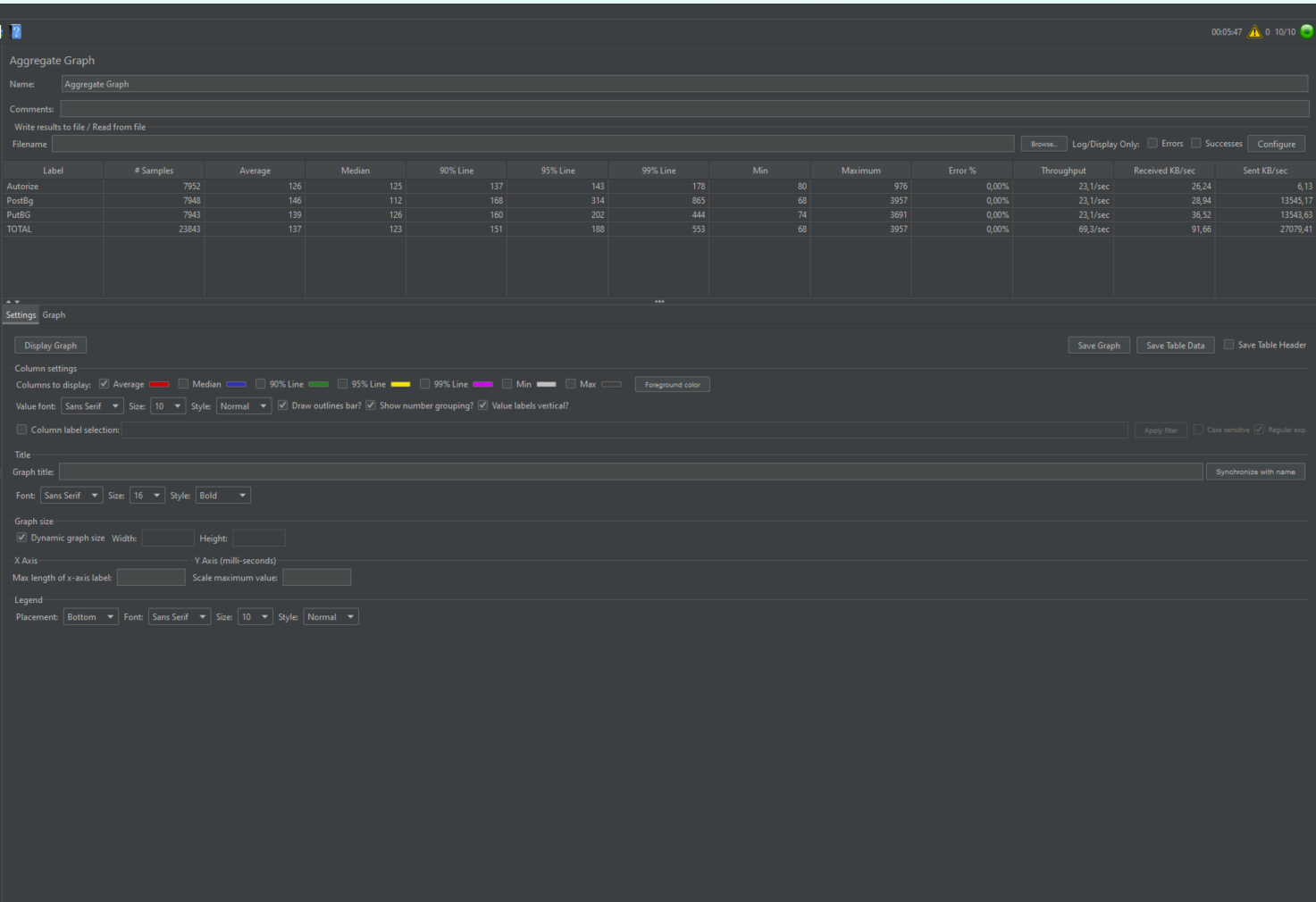


Рисунок 12 Результат при 10 издателях

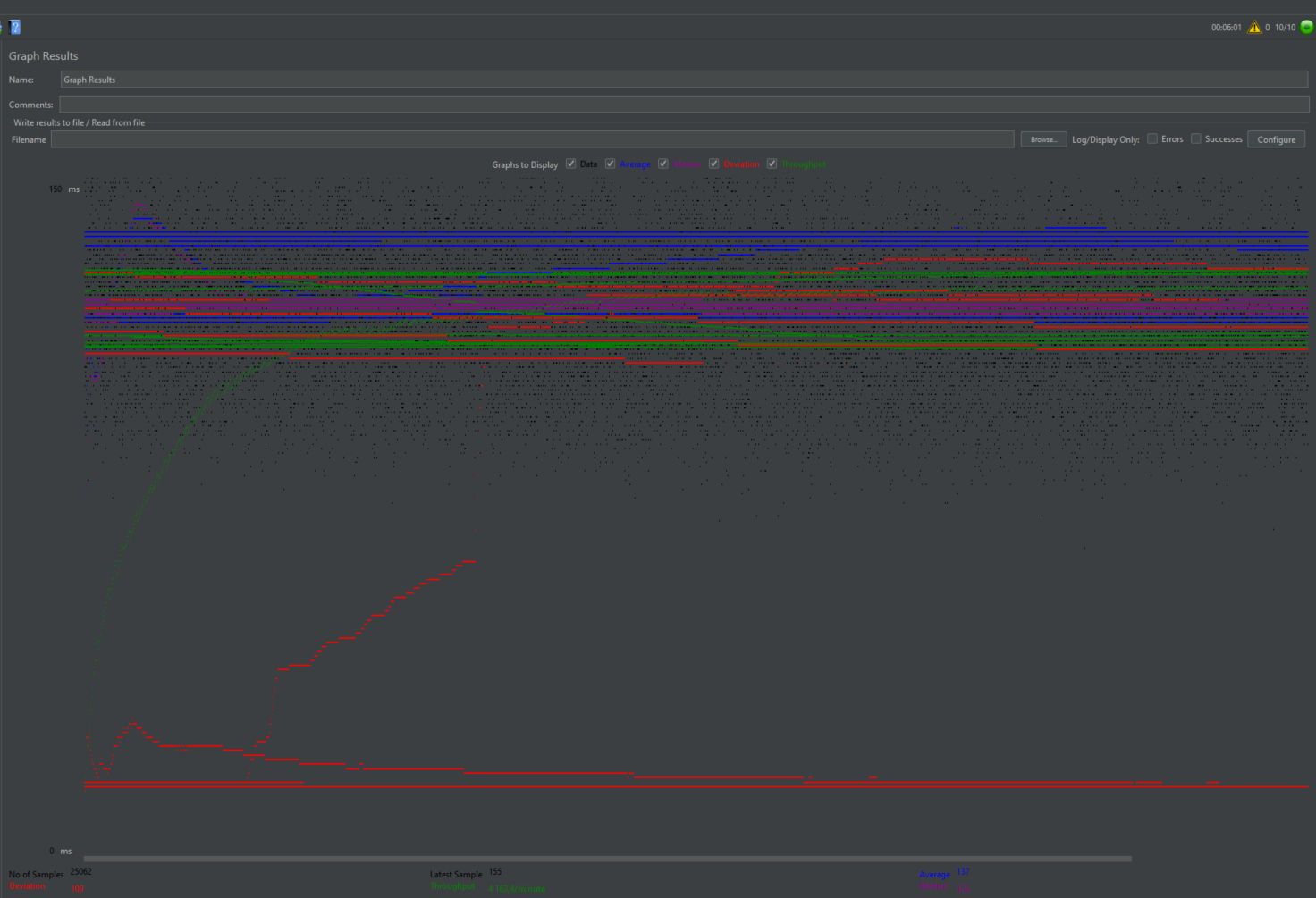


Рисунок 13 Результат при 10 издателях

При 20 издателях система работает тоже стабильно, но 0,06 процента выдает ошибки отсутствия свободных подключений JDBC.Данные ошибки пока не сильно влияет на стабильность системы, так что увеличиваем количество издателей до 40.

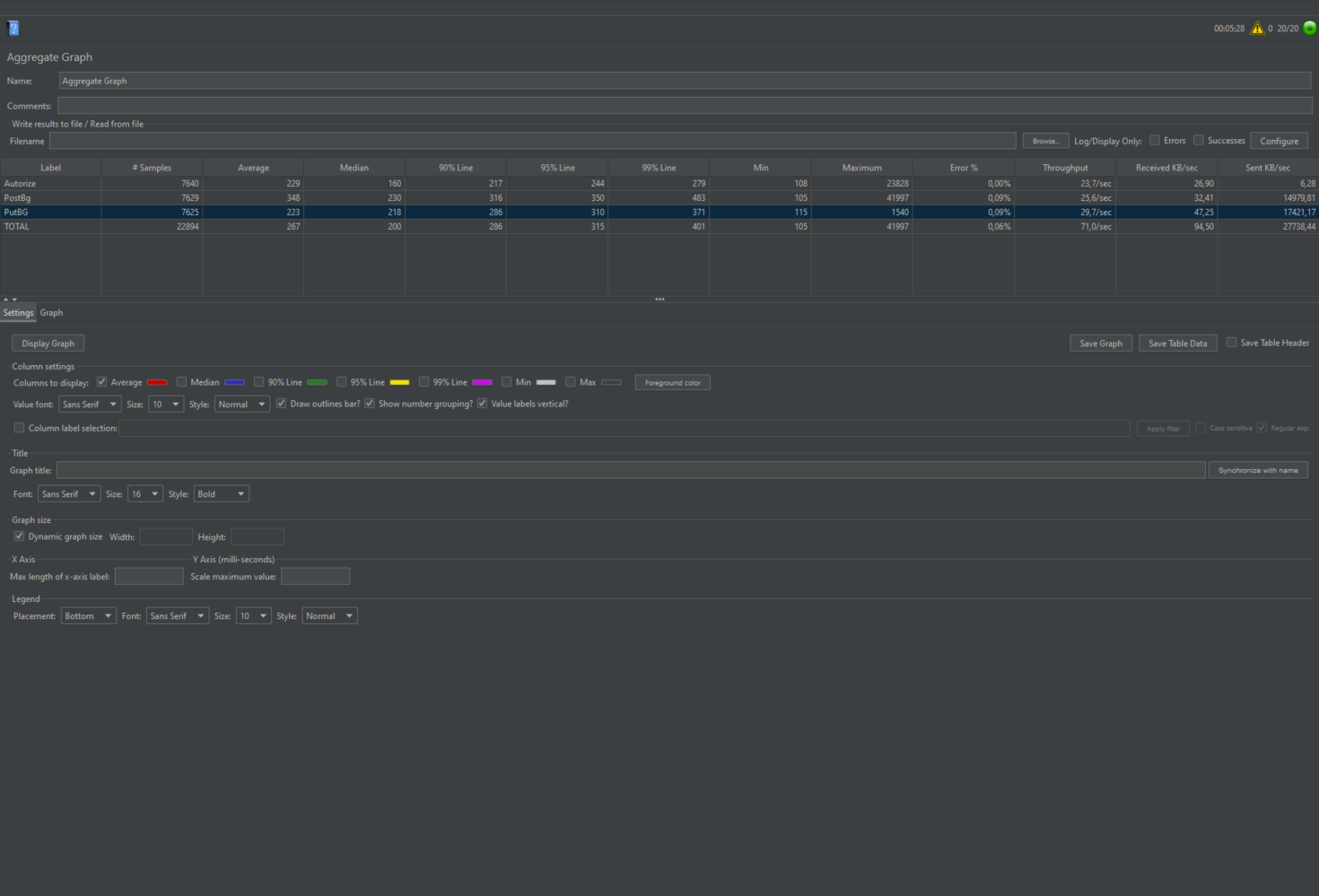


Рисунок 14 Результат при 20 издателях

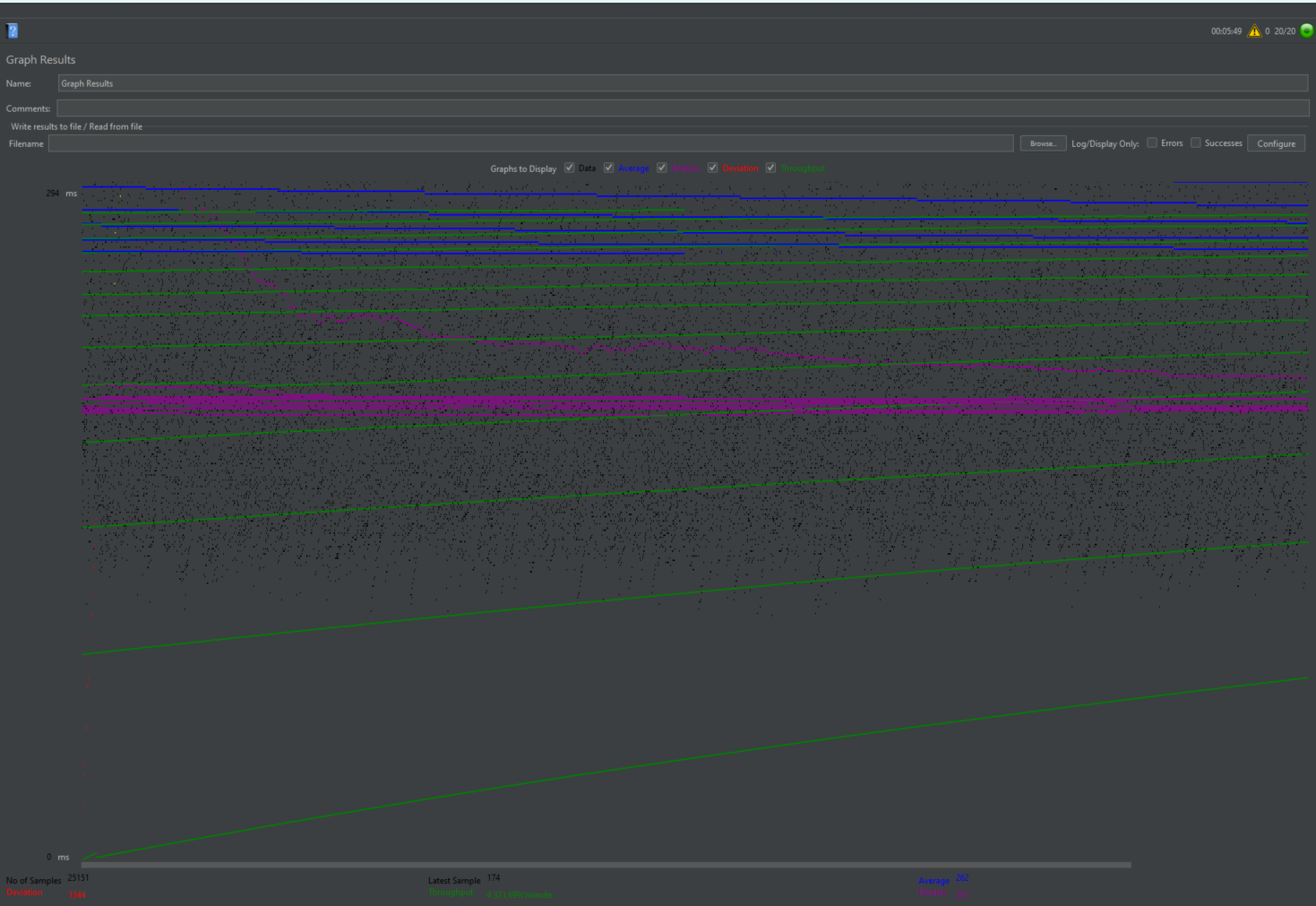


Рисунок 15 Результат при 20 издателях



Рисунок 16 Ошибка отсутствия свободных подключений JDBC

При 40 пользователях система работает достаточно нестабильно, и выдает ошибки на 25 процентах запросов. При ошибке на этапе создания статьи о настольной игре, далее в потоке мы не можем получить идентификатор настольной игры, поэтому выдает ошибка на этапе изменения статьи, но ошибка не влияет на целостность системы. Продолжим тестирование и увеличим количество авторов до 80.

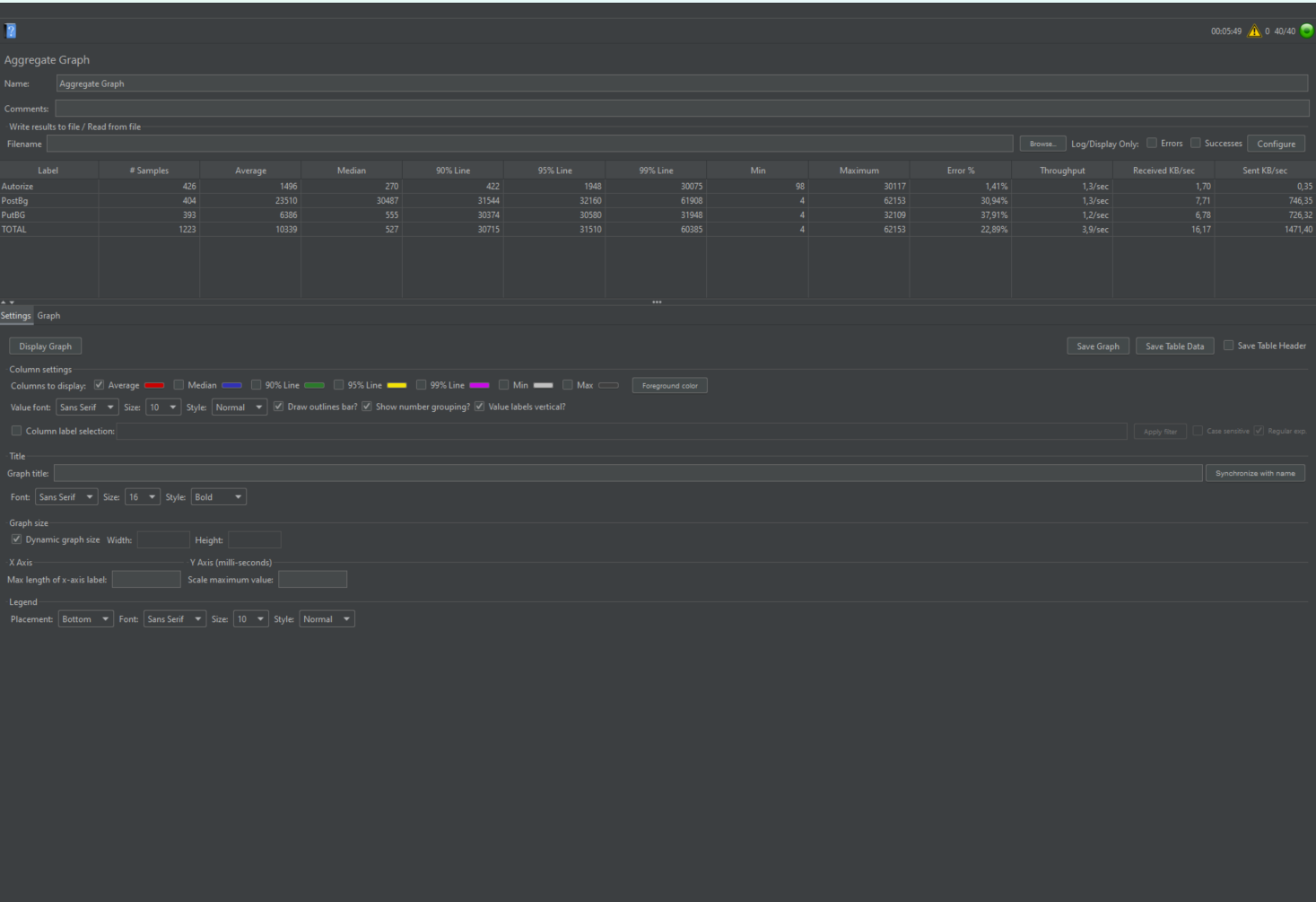


Рисунок 17 Результат при 40 издателях

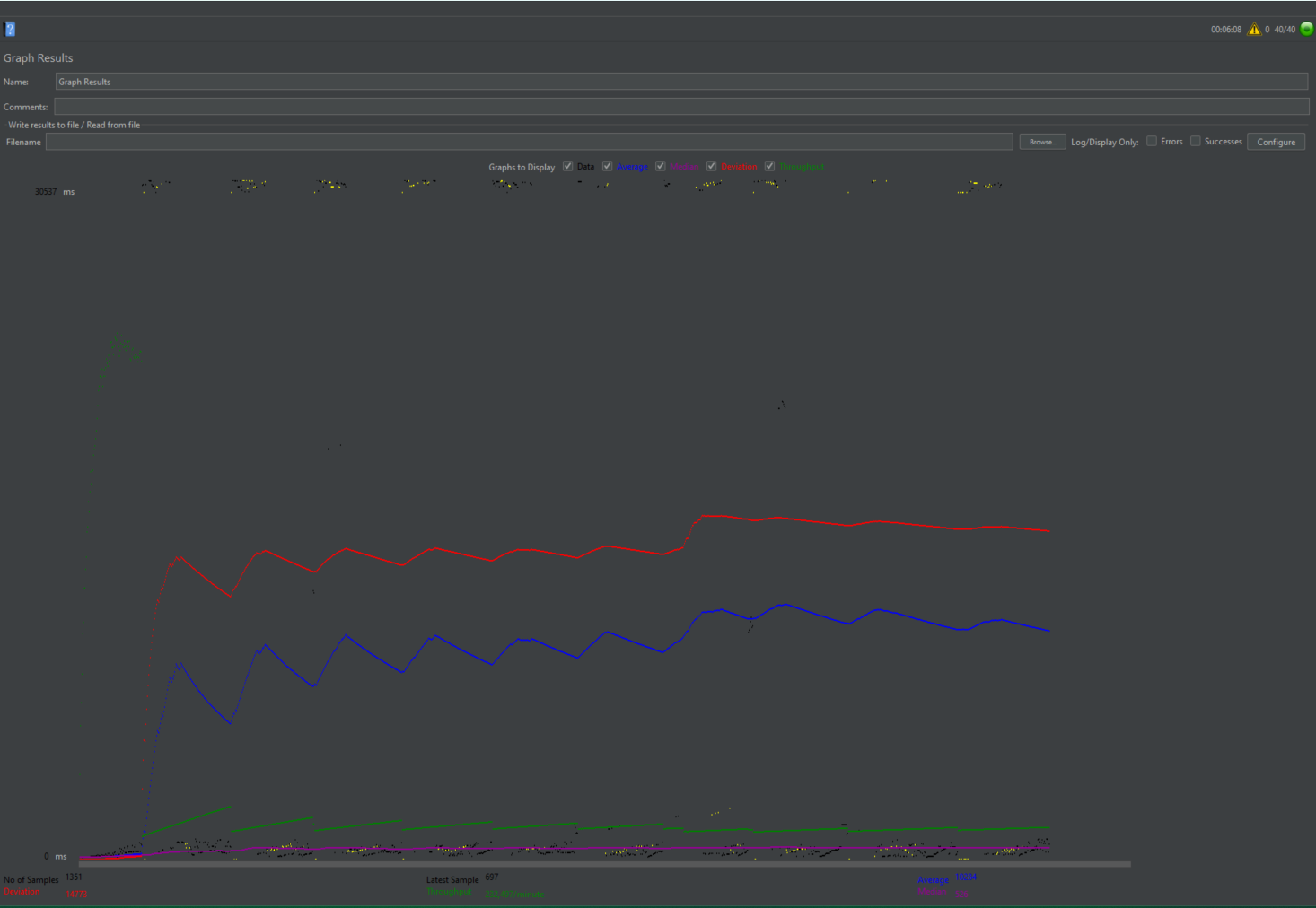


Рисунок 18 Результат при 40 издателях

При 80 издателях теряет свою работоспособность и на запросы о создании статьи отвечает лишь в 40 процентах случаев. Поэтому берем 80 процентов от 80 издателей и запускаем тестирование на 15 минут.

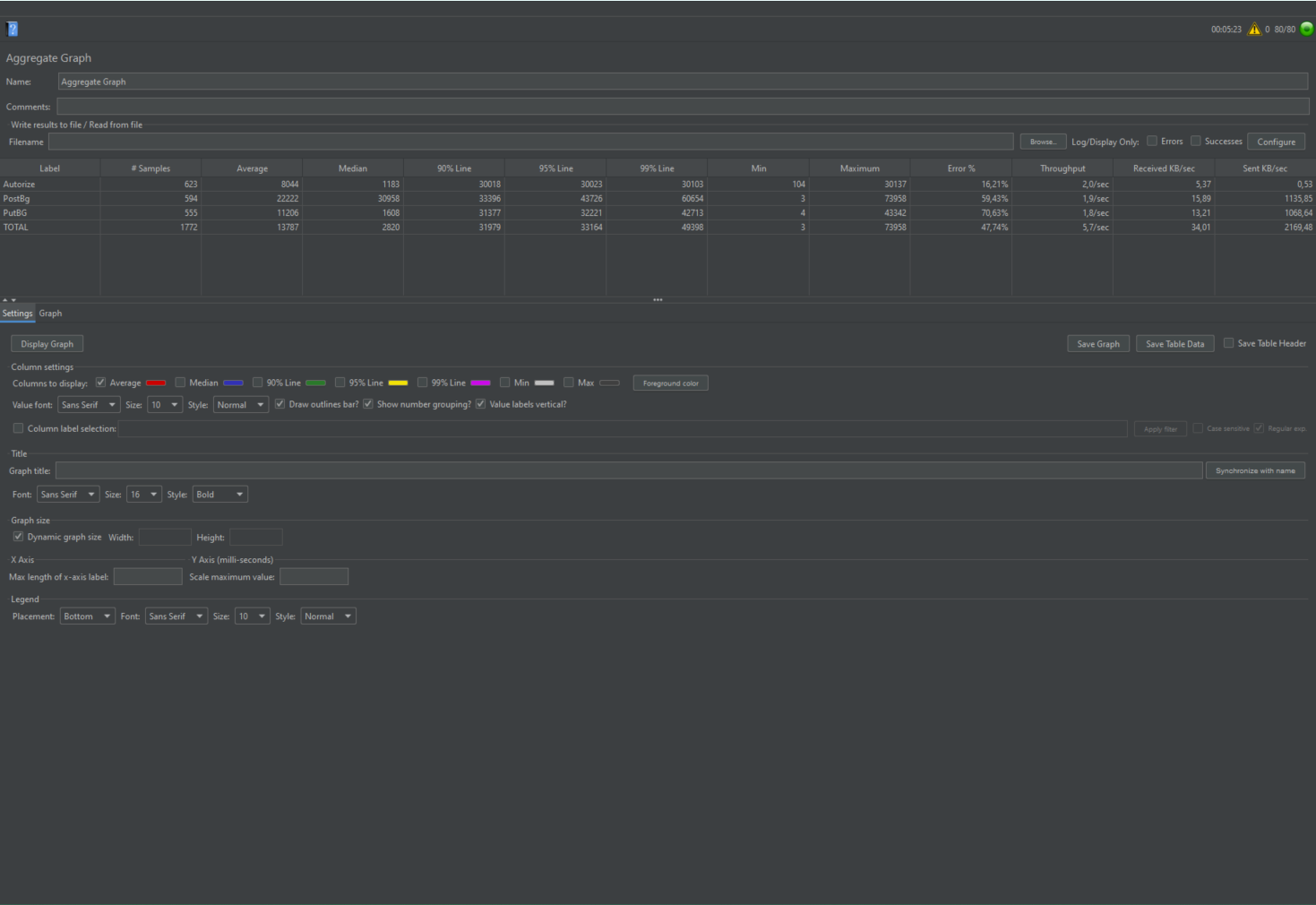


Рисунок 19 Результат при 80 издателях

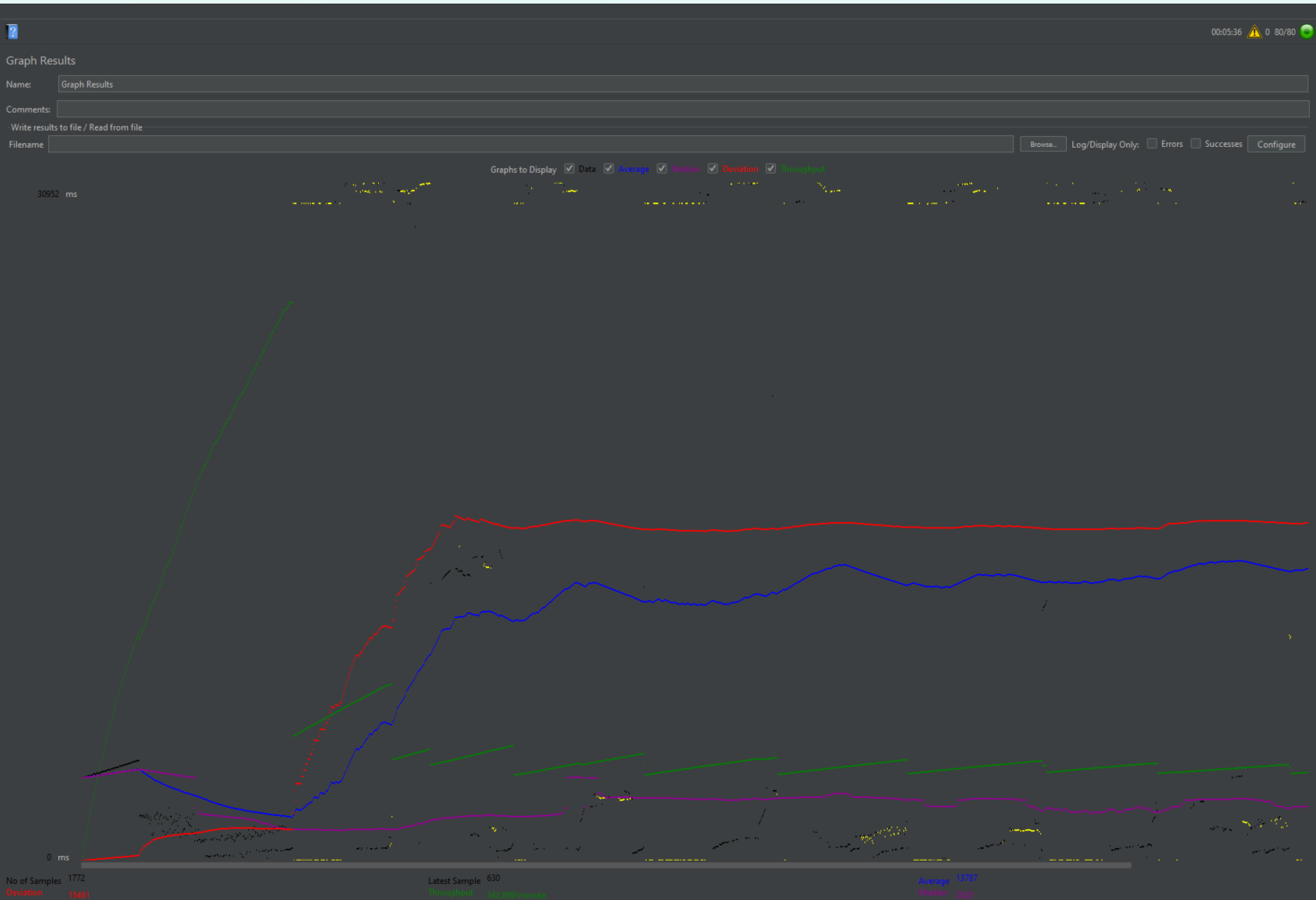


Рисунок 20 Результат при 80 издателях

При 64 пользователях за 15 минут система показала себя более стабильно, чем при 80 пользователях. Она смогла ответить на более 50 процентов запросов.

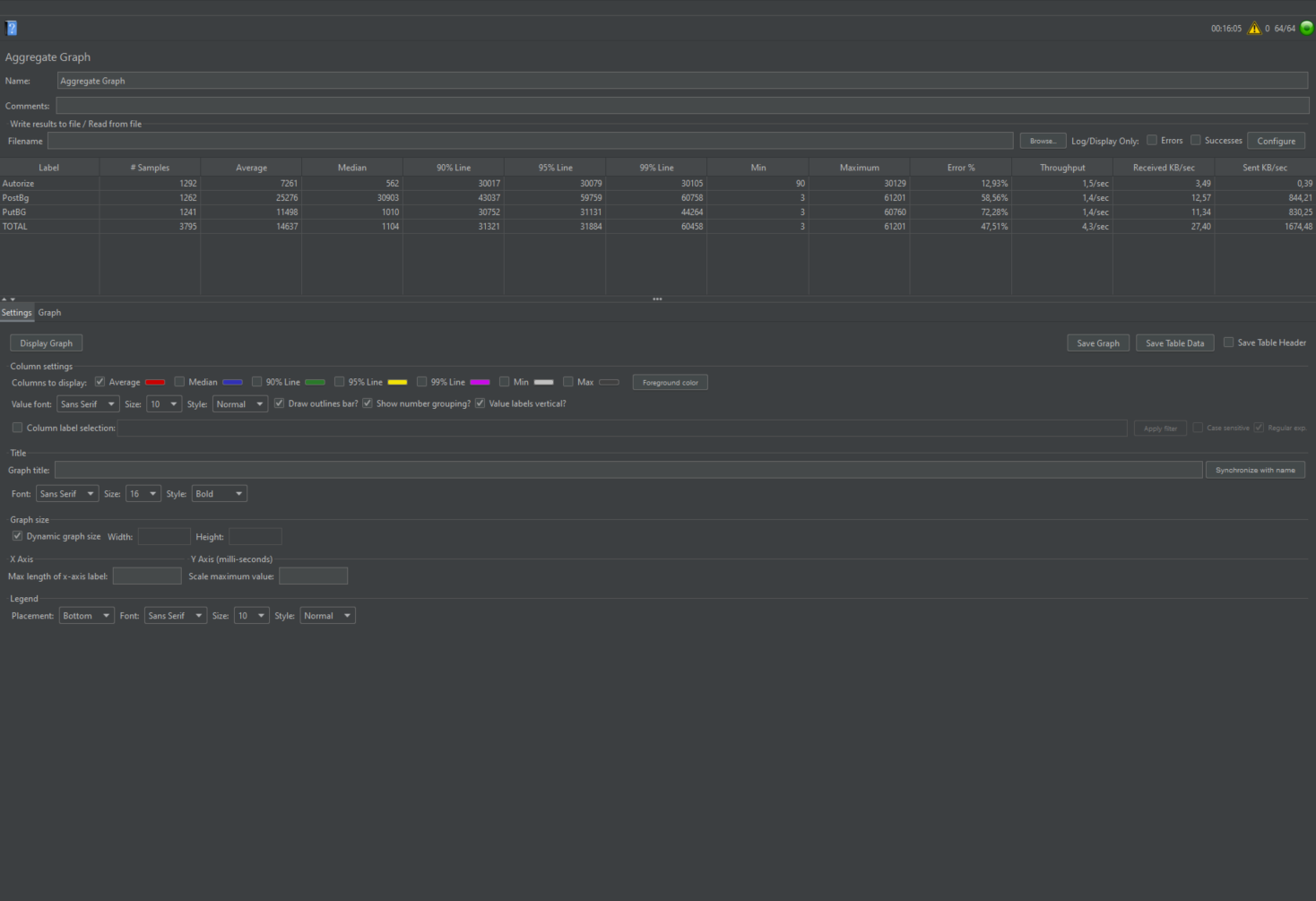


Рисунок 21 Результат при 64 издателях

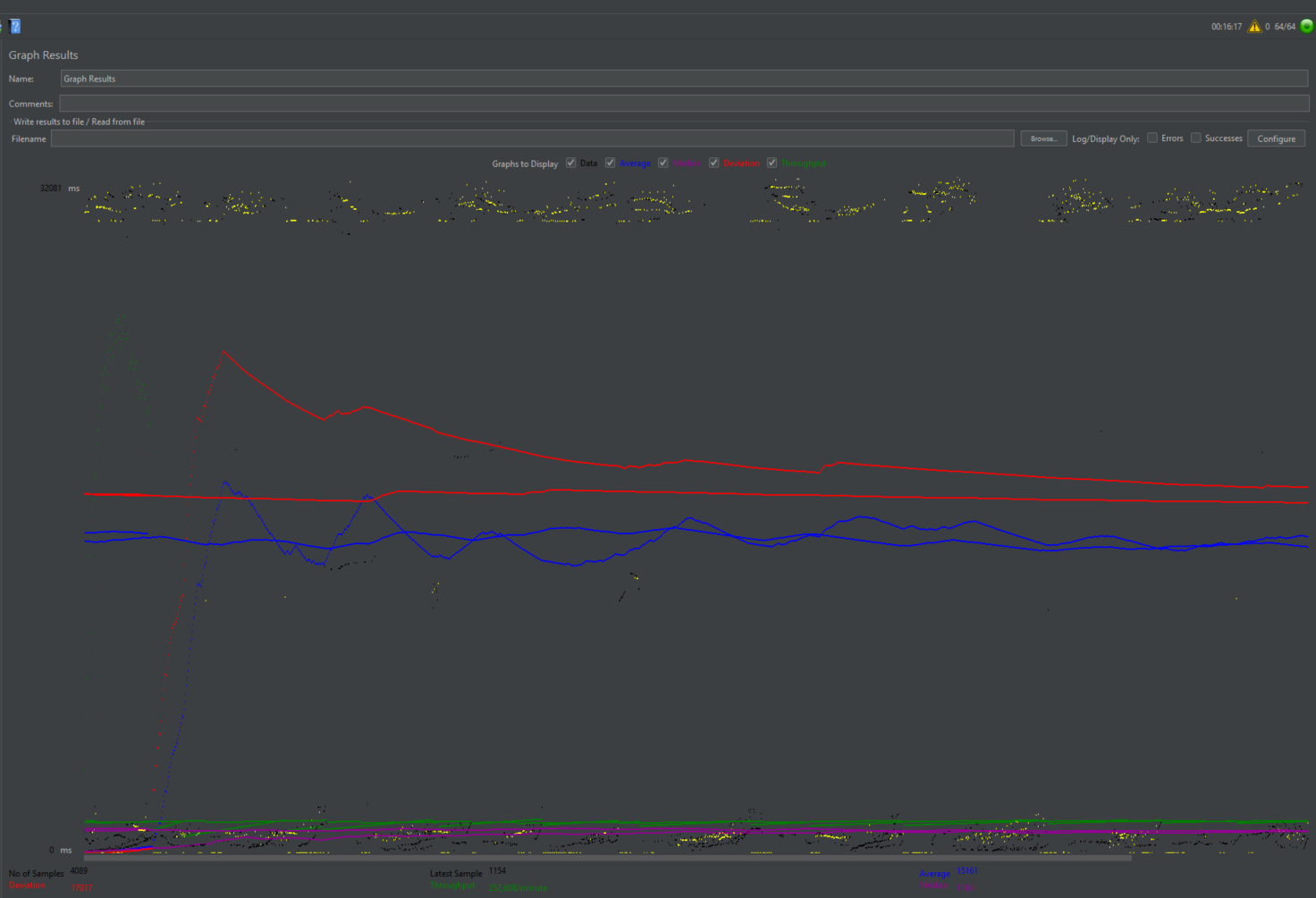


Рисунок 22 Результат при 64 издателях

Игрок

Игрок роль у которой есть больше всего возможностей в системе. Его сценарий состоит из 7 пунктов:

1. Авторизация;
2. Запрос всех настольных игр в системе;
3. Запрос 10 статей о настольных играх;
4. Комментирование 10 настольных игр;
5. Получения списка рекомендуемых настольных игр;
6. Получения списка мероприятий;
7. Вход и выход из 10 случайных мероприятий;

Начнем мы также как и раннее с 10 игроков. При 10 игроках система чувствует стабильно. По этому продолжим на 20 игроках.

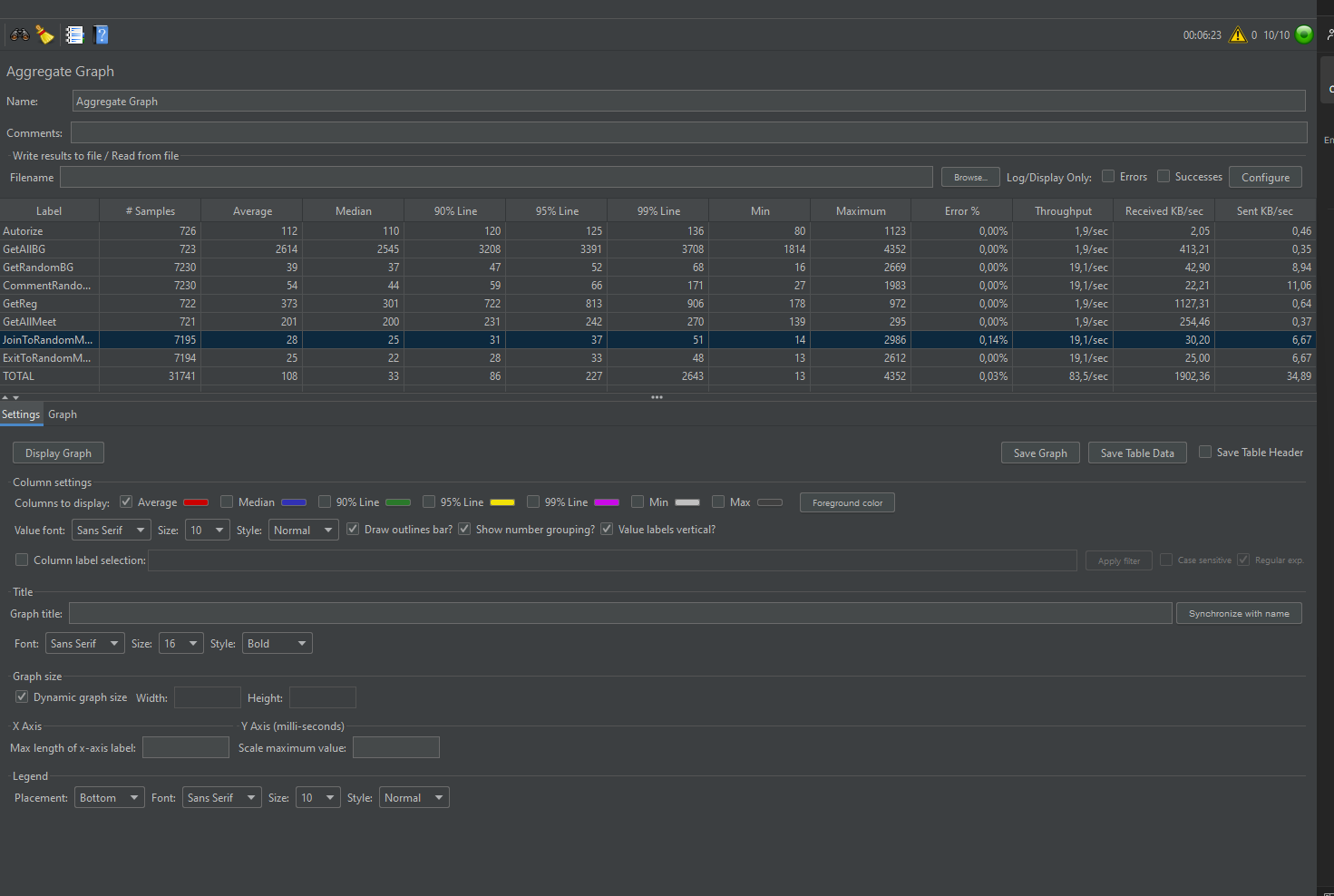


Рисунок 23 Результат при 10 игроках



Рисунок 24 Результат при 10 игроках

При 20 игроках система так же чувствует себя стабильно. Увеличиваем количество игроков до 20.

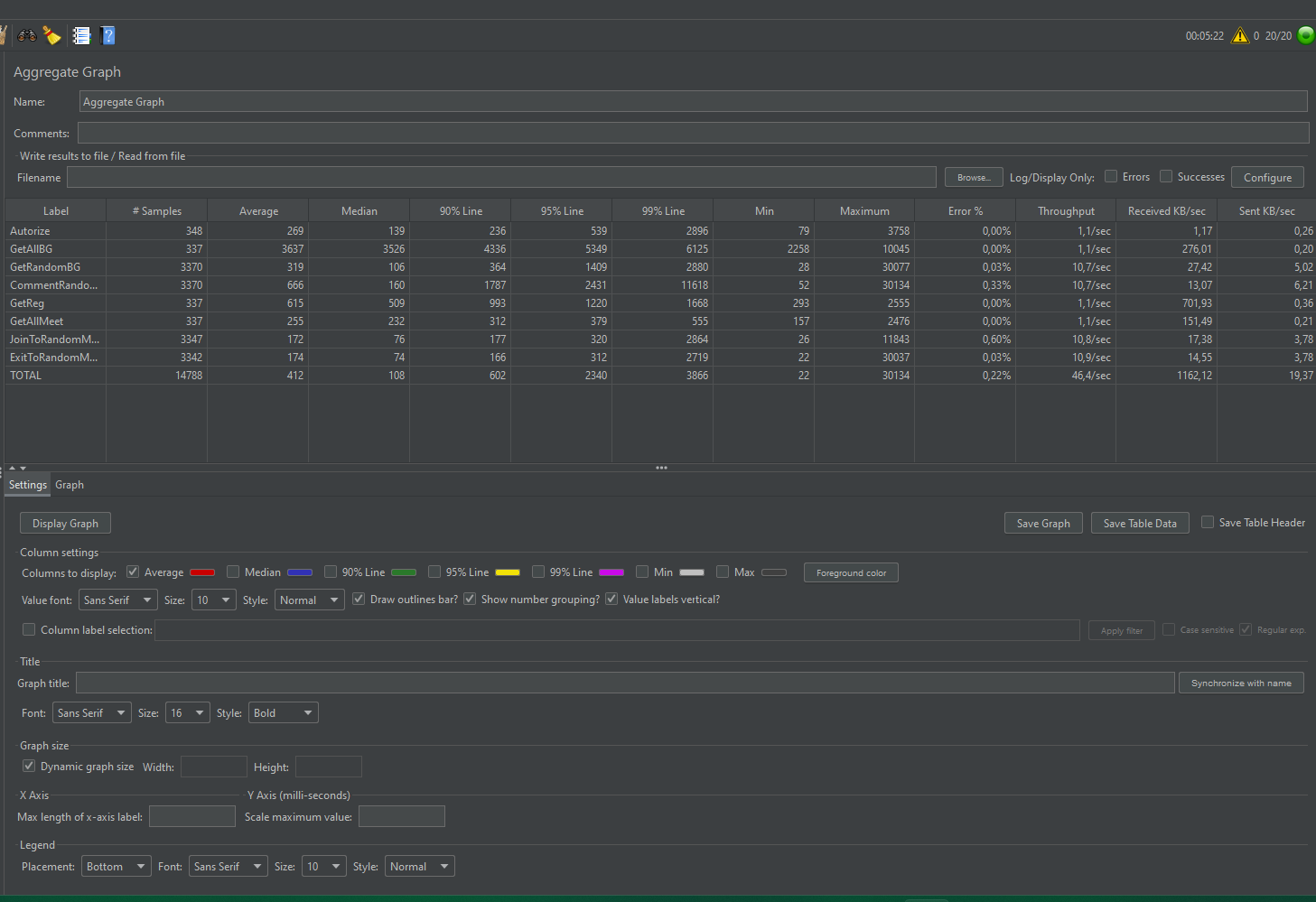


Рисунок 25 Результат при 20 игроках

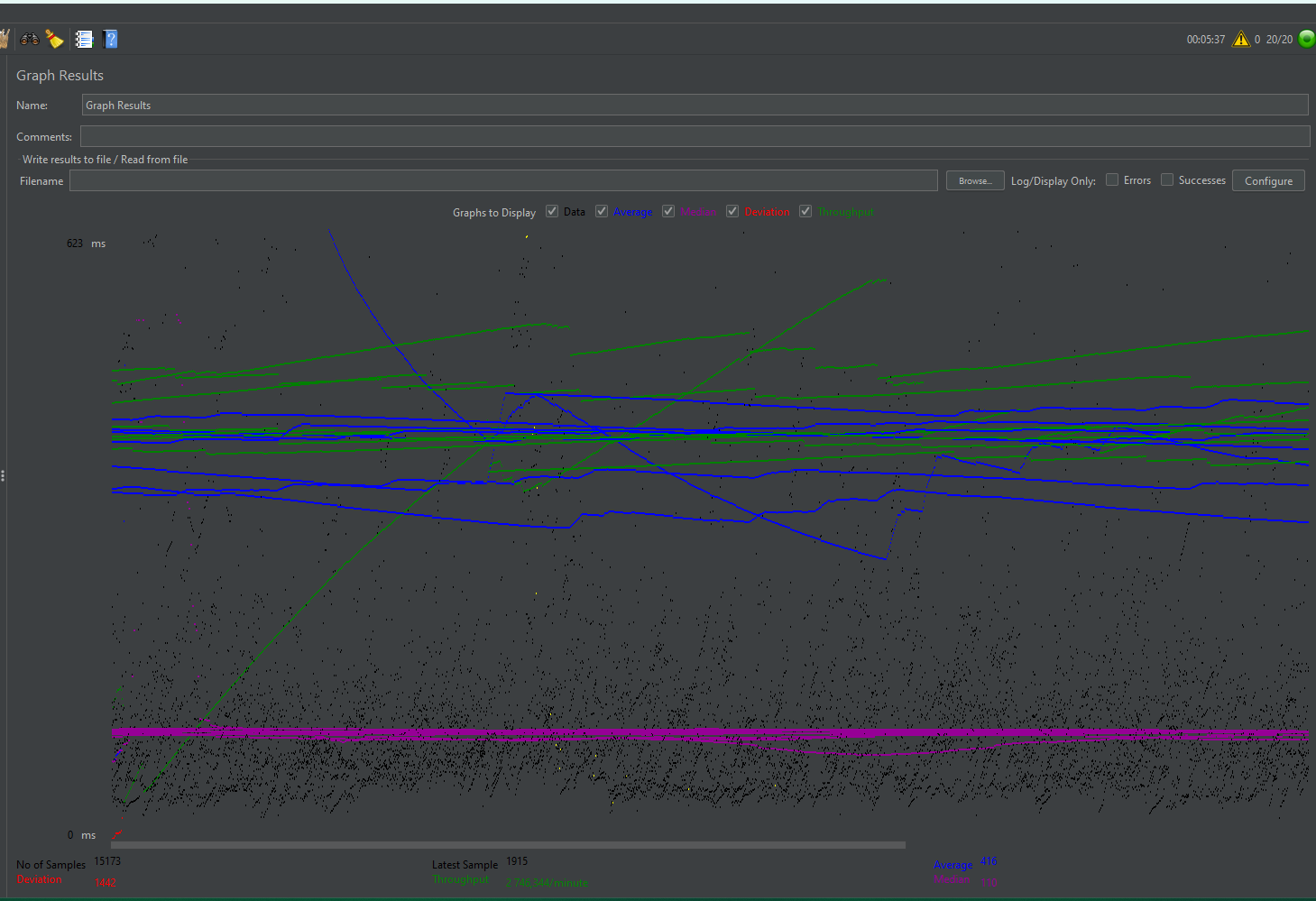


Рисунок 26 Результат при 20 игроках

При 40 игроках система чувствует себя не стабильно и игнорирует 90 процентов запросов. Поэтому берем 32 пользователя и тестируем их 15 минут.

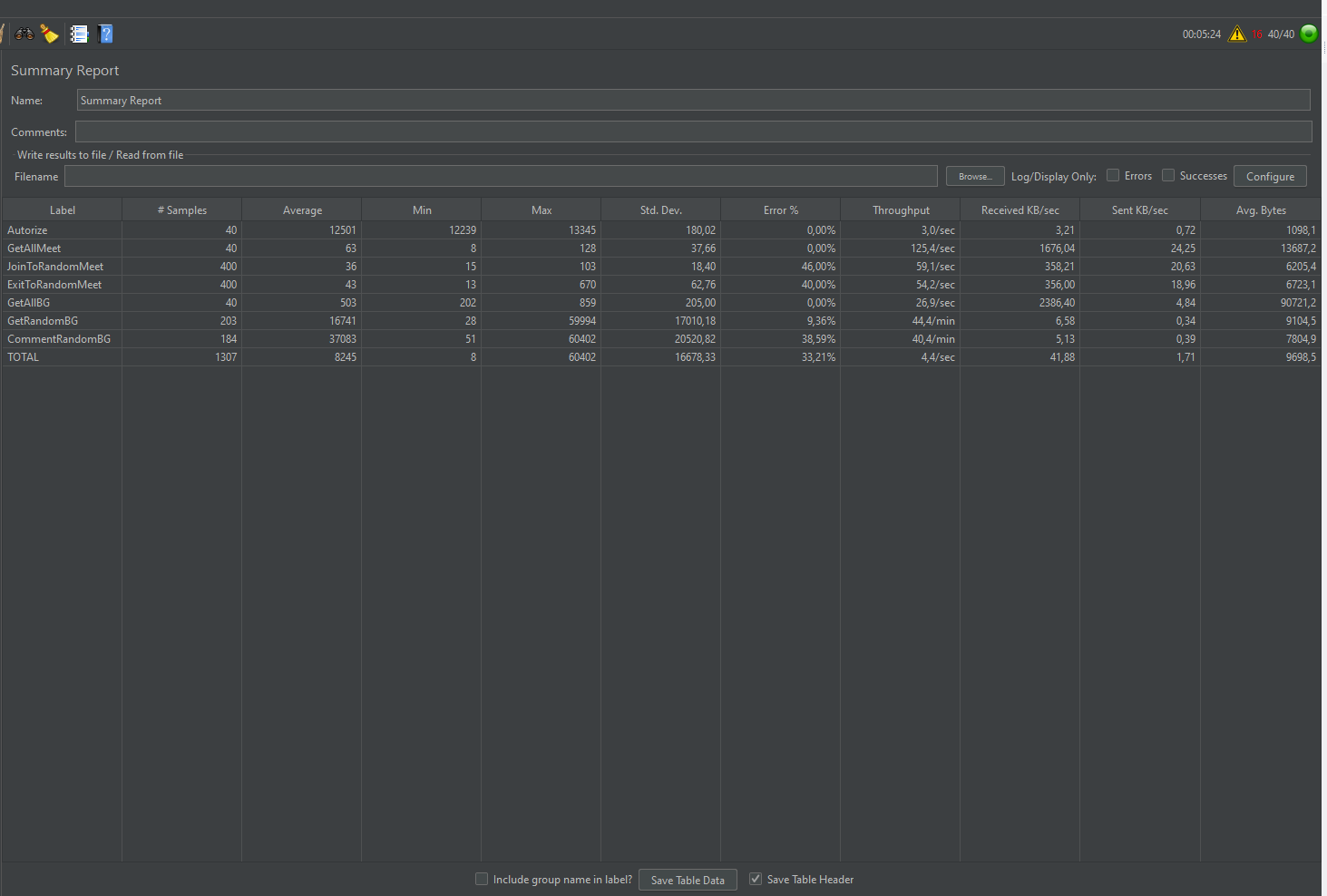


Рисунок 27 Система при 40 игроках

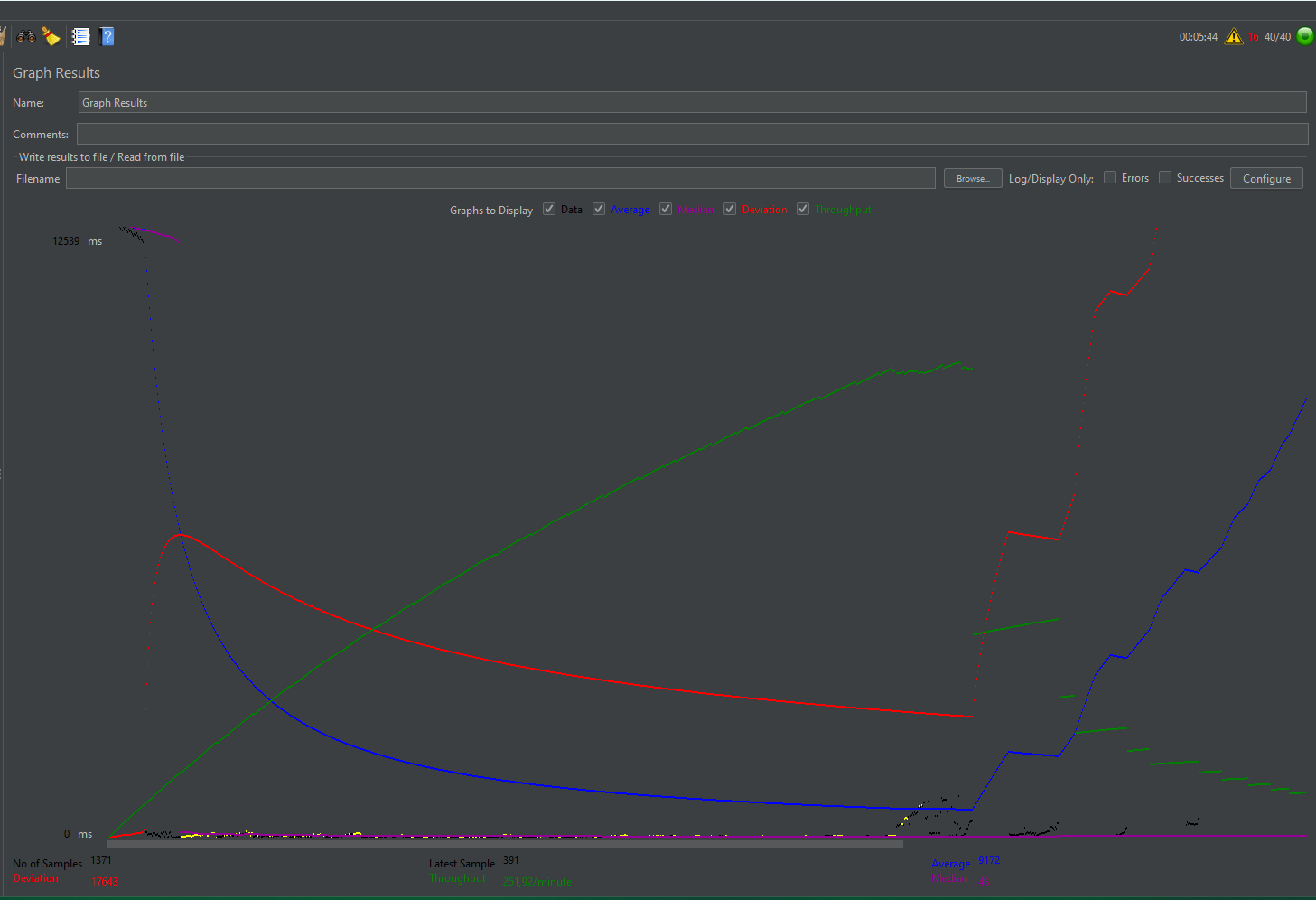


Рисунок 28 Система при 40 игроках

При 32 пользователях работает стабильнее.

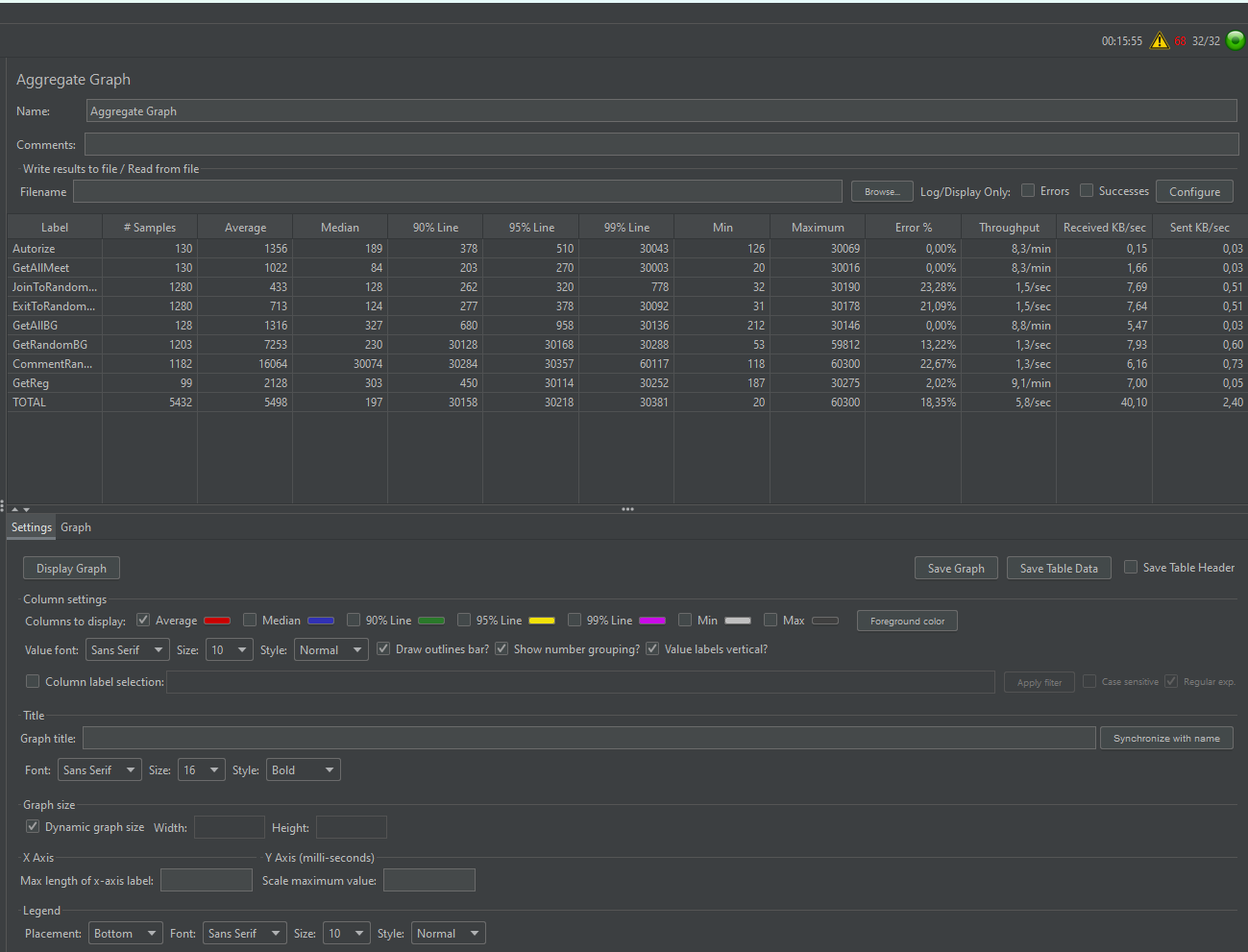


Рисунок 29 Результат при 32 пользователях

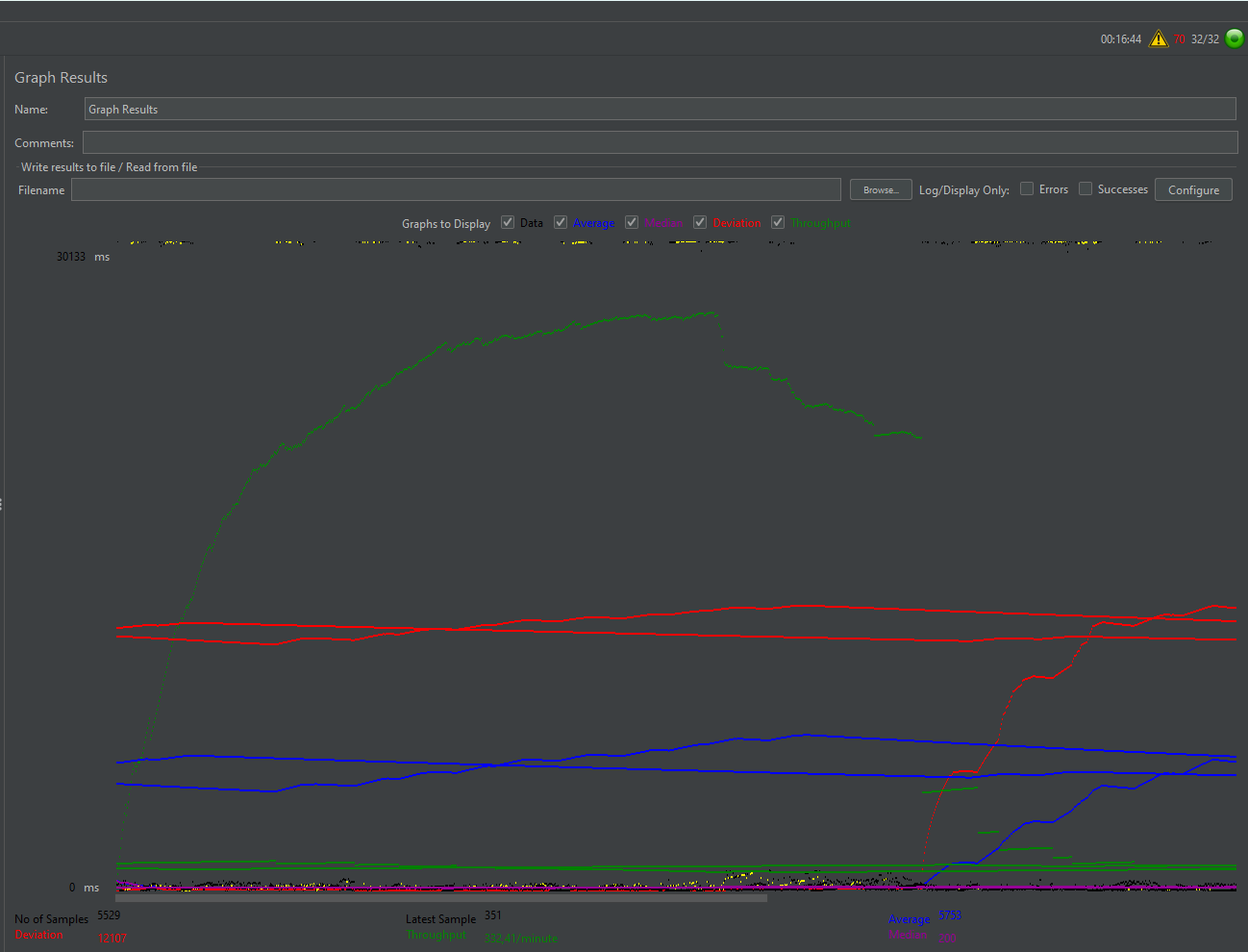


Рисунок 30 Результат при 32 пользователях

Вывод нагрузочного тестирования

Систем была протестирована и были выявлены границы количества пользователей системы одновременно. Система стабильно выдерживает 20 игроков,20 издателей и 40 организаторов.