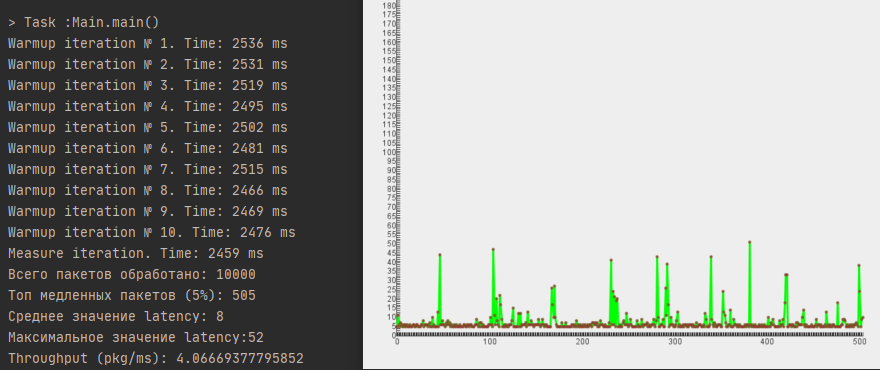
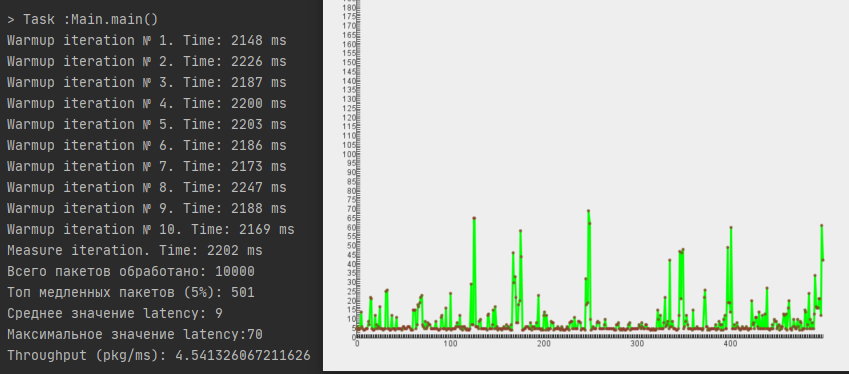
В исследовании проводились прогревочные запуски для наиболее точных результатов измерений. В качестве показателей использовались latency (среднее, максимальное и latency 5% самых медленных пакетов (график)), throughput. Значение latency вычислялось как разница времени подачи пакета в сеть и выхода его на нужном узле. Затем вычислялись 5% самых медленных пакетов и рисовался график. Throughput вычислялся как количество пакетов прошедшее за 1 ms за время работы программы после ее выполнения.

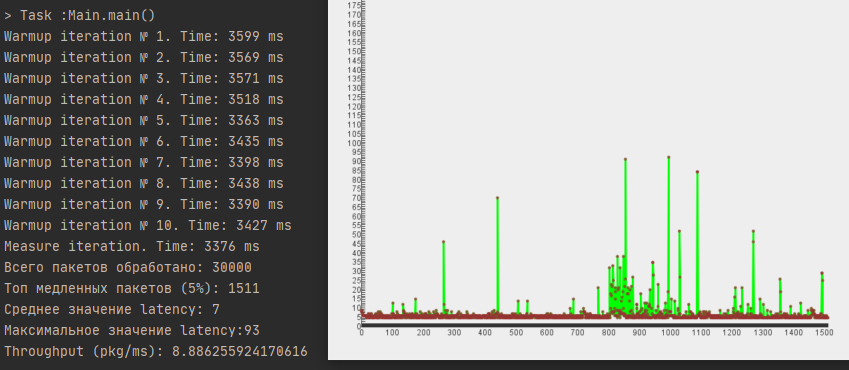


Минимальная нагрузка на сеть. Есть пакеты, которые обрабатывались достаточно долго (скорее всего несколько пакетов попали на один узел), в среднем же обработка достаточно быстрая.

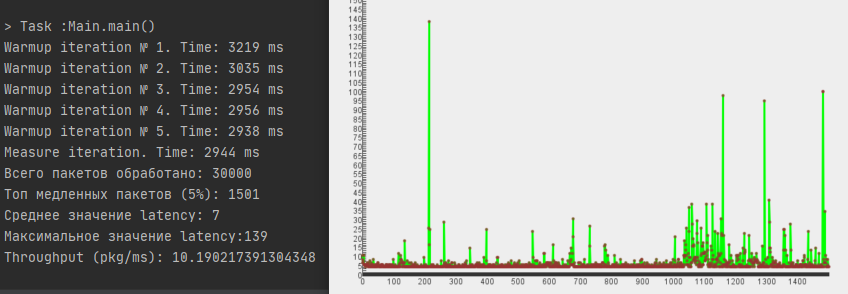


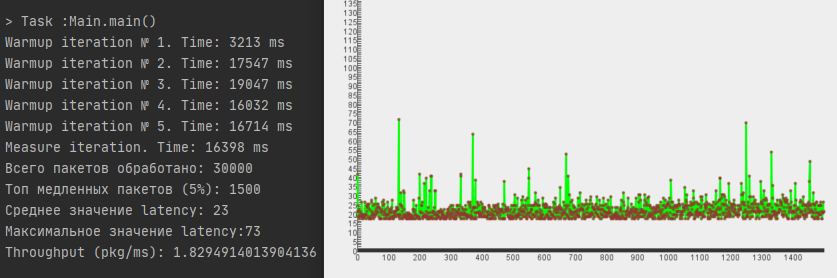
Более высокая нагрузка на сеть (х10). Как мы видим уже больше пакетов начинают пересекаться, в связи с чем, количество медленных пакетов возросло. Throughput при этом вырос, но не очень значительно. Среднее значение latency тоже возросло, т.к. увеличились пересечения в узлах.

Далее проверим что получится, когда мы будем грузить только 2 соседние ноды.

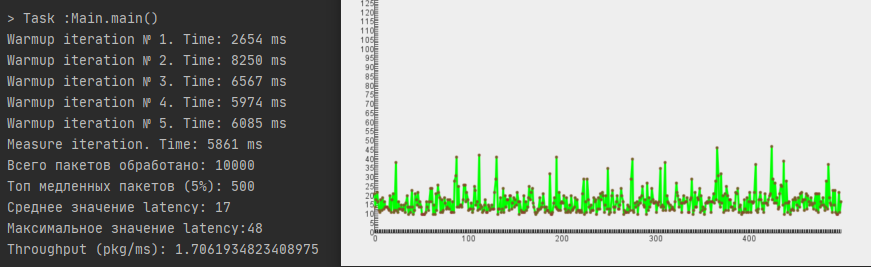


Как мы видим, ближе к середине теста происходят накопления, но не сильно критичные. Throughput при этом растет очень сильно. Latency лучше чем в предыдущем тесте – где мы грузили 5 разных (рандомных) нод.

Повысим нагрузку и будем грузить 3 подряд идущих ноды, чтобы увеличить на них пересечения.   
  
Как видим к концу теста появились значительные просадки в latency. Пики уже значительно выделяются на фоне остальных пакетов. При этом Throughput стал ещё больше. Среднее же latency не поменялось.   
Увеличиваем нагрузку. Грузим 4 последовательных ноды.



Как видим кольцо забилось и уже не может эффективно работать. Т.е. начиная с какого-то числа пользователей кольцо перестает показывать хорошие результаты. Видим, что throughput стал совсем низким – 1.8 – это ниже чем в первом тесте (низкая нагрузка). Да, нет очень больших просадок по latency, но при этом среднее latency выросло почти втрое, что очень существенно.   
Попробуем провести похожие измерения, но уже с размером очереди не 5, а 2 элемента.



Как видим, кольцо забивается уже на самом первом тесте (с минимальной нагрузкой). Throughput очень маленький, latency – большое – почти в 7 раз больше чем при начальных условиях.  
Скорее всего есть некая неточность в измерениях. Необходимо было поставить ещё один класс(контейнер) перед tokenRing, в который приходили бы пакеты (т.е. реальное время прихода клиента в наш сервис) Сейчас же мы засекаем время прихода клиента в момент, когда он вошел в tokenRing. Это не совсем правильно, т.к. при забитом кольце клиент сидит и ждет очереди войти. Но даже при таких неточных измерениях мы можем видеть, что у кольца есть свой предел. Таким образом при каком-то количестве клиентов (количество зависит от количества нод, от распределения приходящих клиентов на эти ноды и размеров очереди. При этом большие размеры очереди также не выход – т.к. очень сильно будет забиваться память)