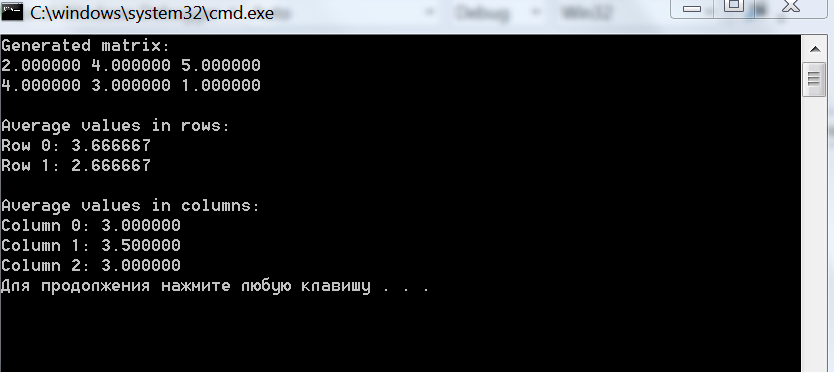
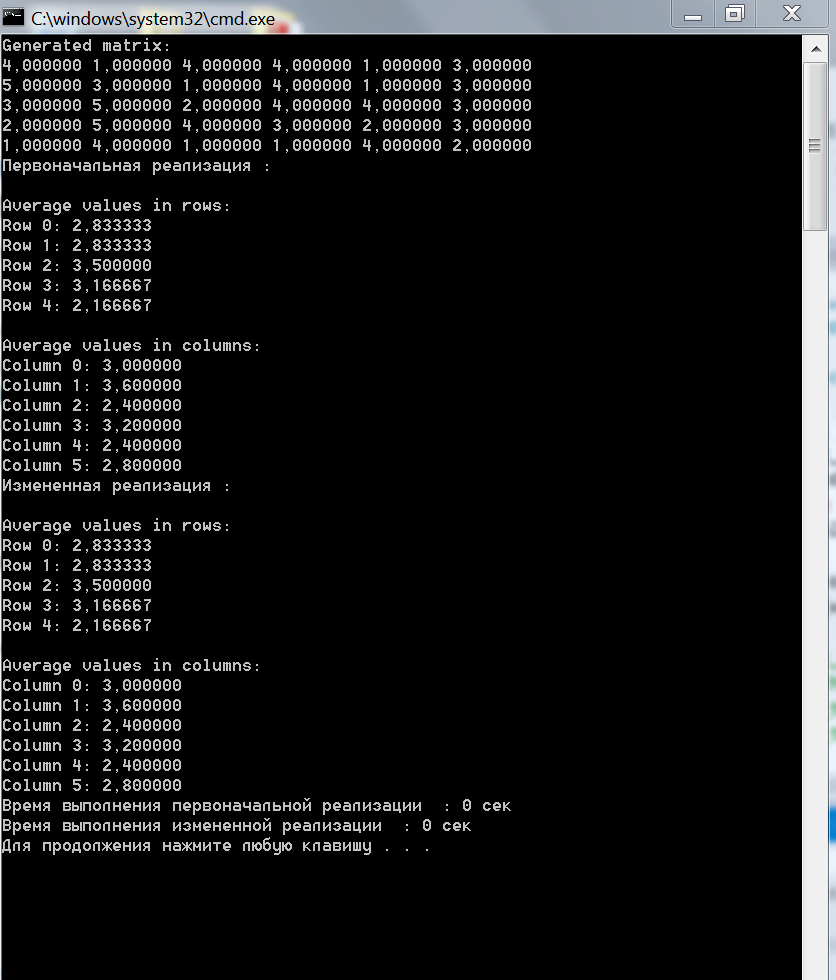
1. Разберите программу представленную в файле [task\_for\_lecture5.cppПредварительный просмотр документаПросмотр в новом окне](https://canvas.instructure.com/courses/1207882/files/56839533/download?verifier=mNdIoUBY4pBi1jde9Am8rkuXWtbga0lU3MLhk9WU&wrap=1). В программе создается 2 потока, каждый из которых вычисляет средние значения матрицы, один по строкам исходной матрицы ***matrix***, а другой - по столбцам. Запустите программу и убедитесь в ее работоспособности.

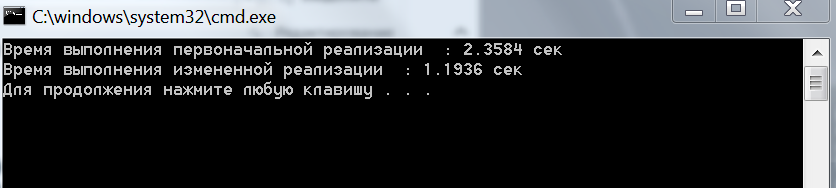


1. Проанализируйте программу и введите в нее изменения, которые по Вашему мнению повысят ее производительность.

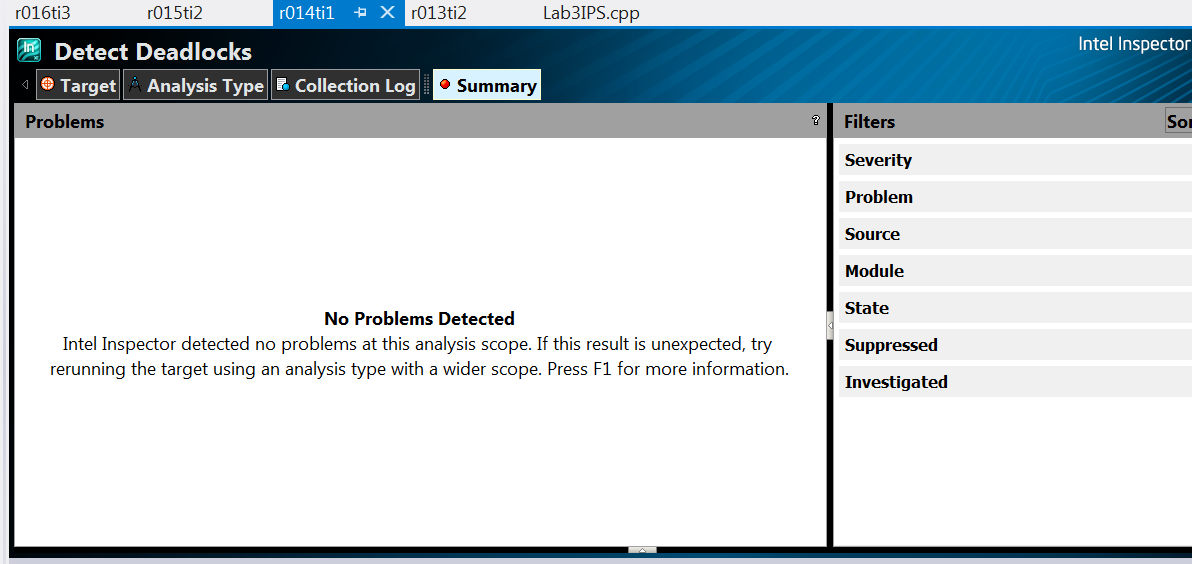
Решения измененной и первоначальной реализации совпали

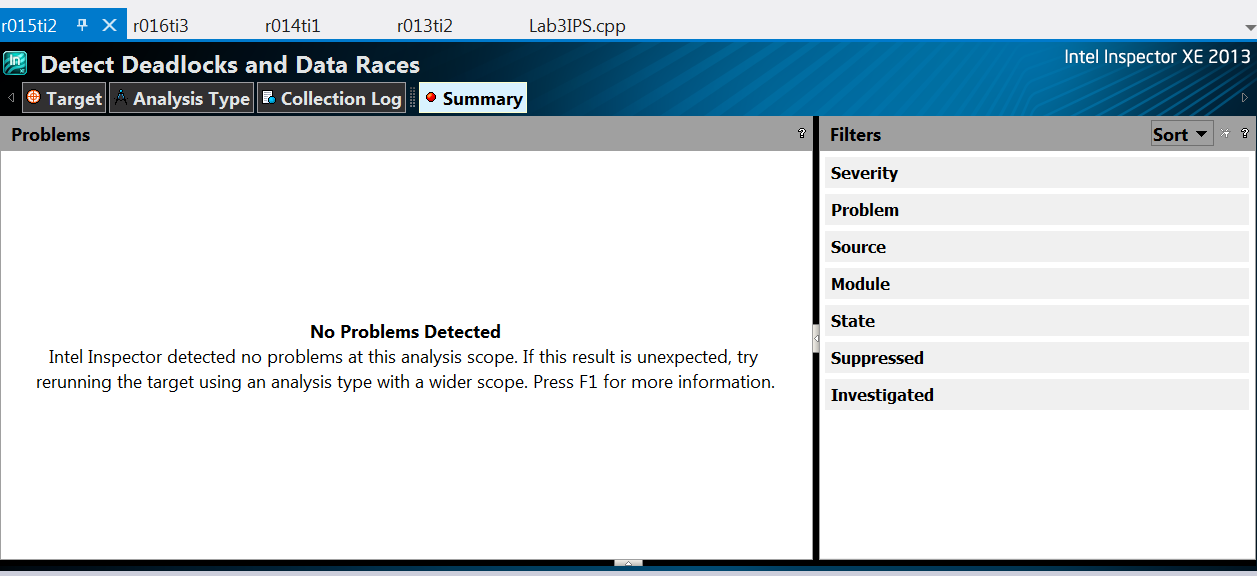


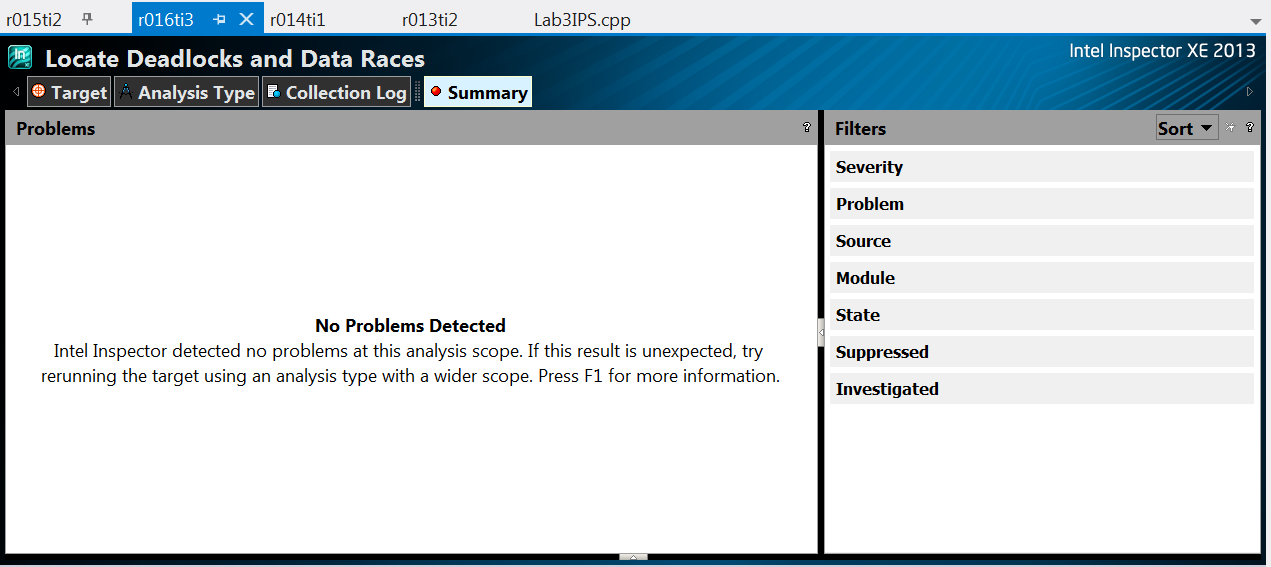
При большой матрице (15000 на 15000 , при больших размерах выдает ошибку ) измененная реализация оказывается быстрей ( если взять четыре потока )



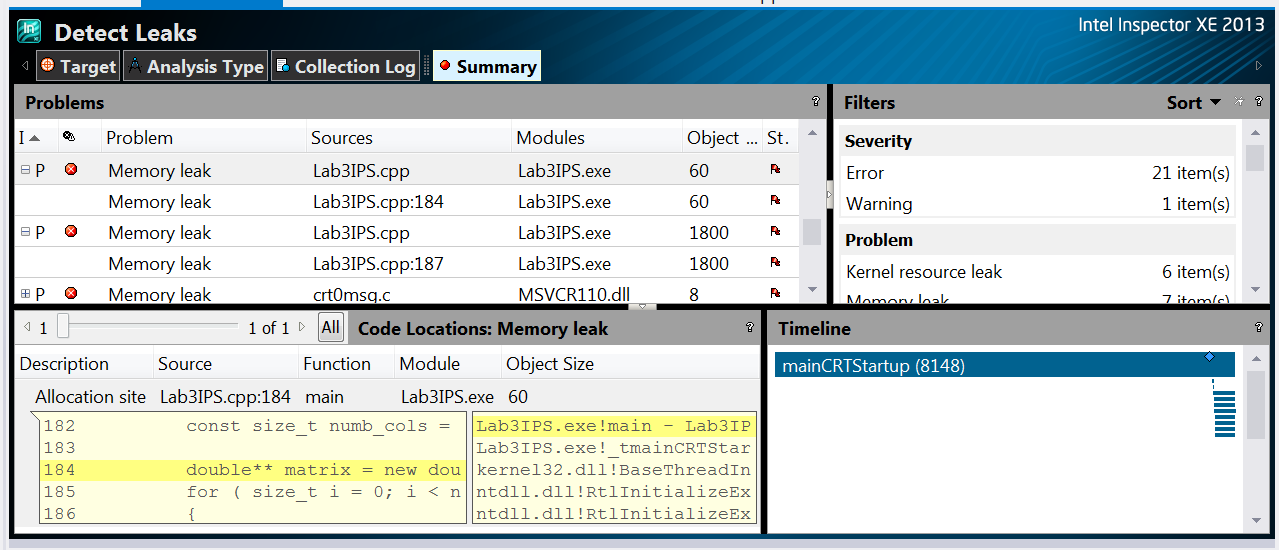
Определите с помощью ***Intel Parallel Inspector*** наличие в программе таких ошибок как: *взаимная блокировка*, *гонка данных*, *утечка памяти*. Сделайте скрины результатов анализа ***Parallel Inspector***(вкладки ***Summary***, ***Bottom-up***) для всех упомянутых ошибок*,*где отображаются обнаруженные ошибки, либо отражается их отсутствие. Запускайте анализы на разных уровнях (***Narrowest***, ***Medium***, ***Widest***).

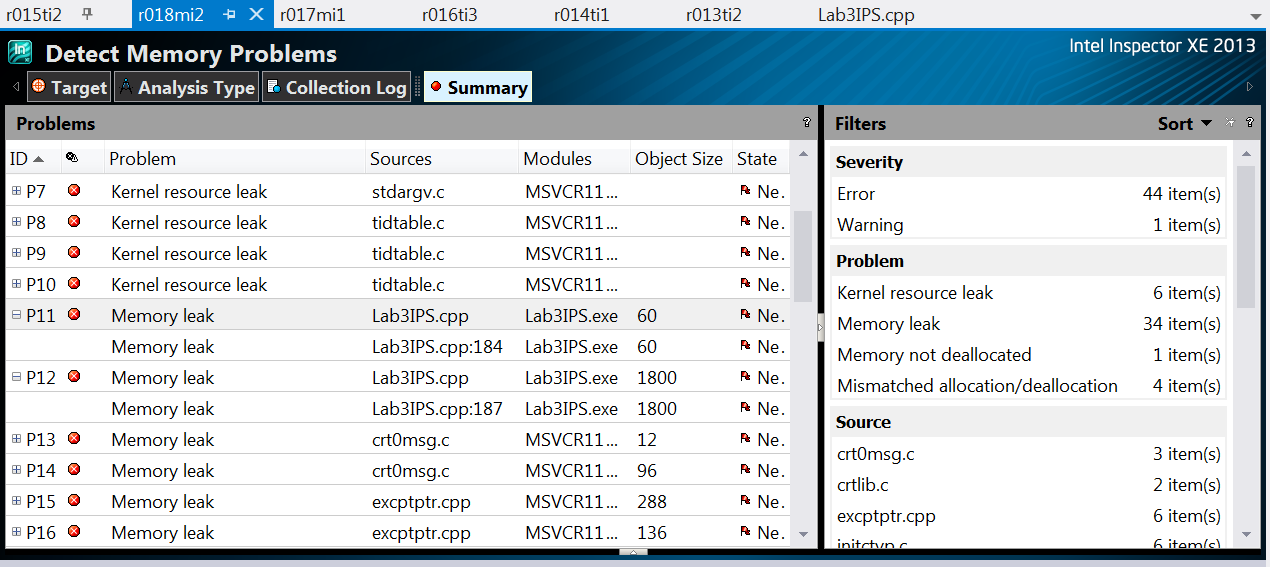


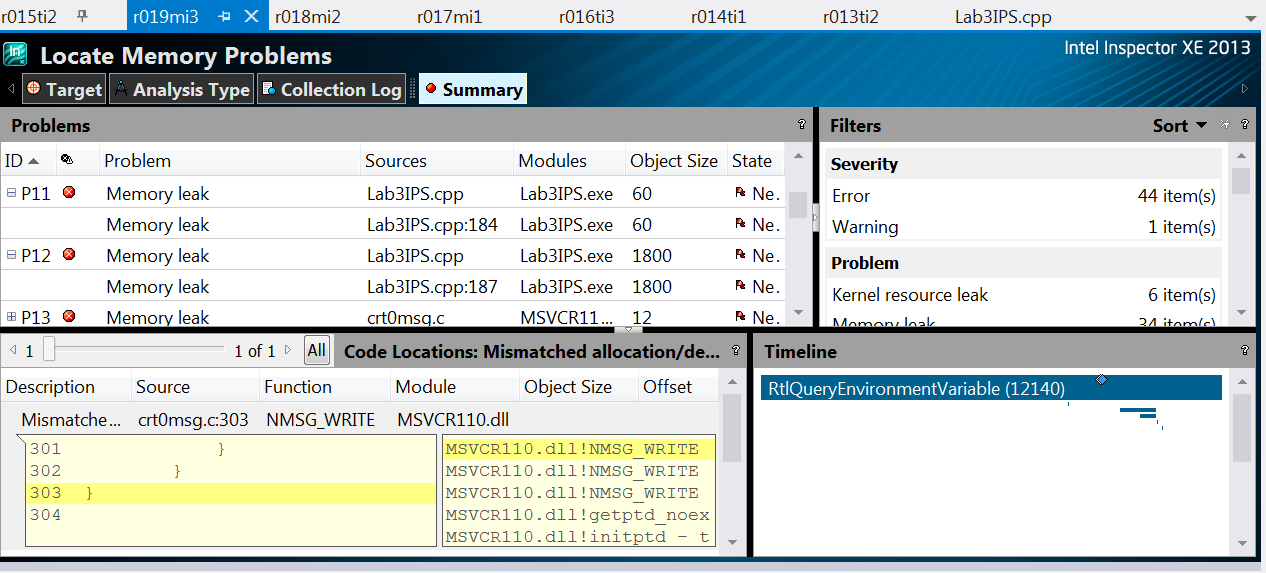




Гонок данных и взаимных блокировок в программе нет







Измените код программы таким образом, чтобы ***Inspector*** при проверке не находил в программе ошибок, перечисленных в **п. 3.**Сделайте скрины результатов запуска***Parallel Inspector***.

Я закомментировал предыдущую реализацию программы , а также добавил строчки удаления динамических массивов :

delete [] average\_vals\_in\_rows;

delete [] average\_vals\_in\_cols;

for (int i = 0; i < numb\_rows; ++i )

{

delete[]matrix[i];

}

delete[] matrix;

Inspector стал выдавать следующие результаты :

