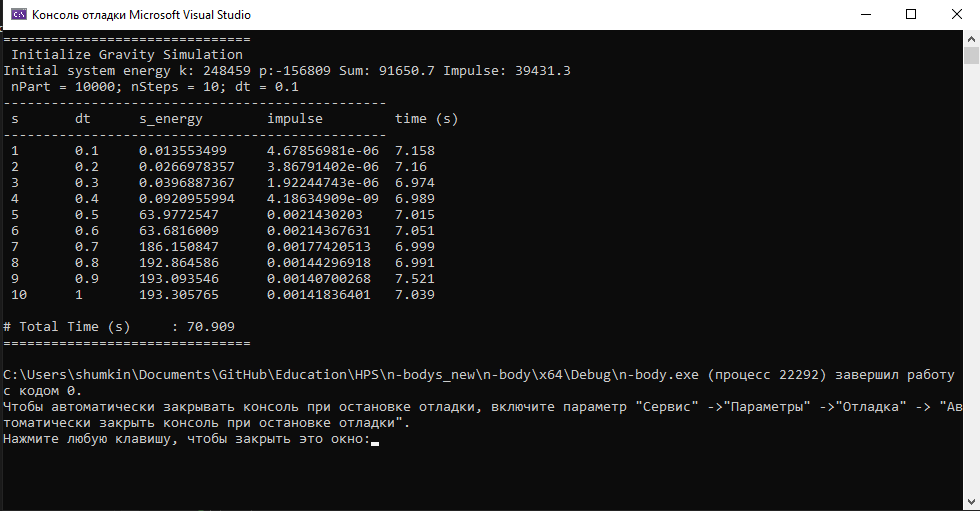
Отчет по гетерогенным вычислениям



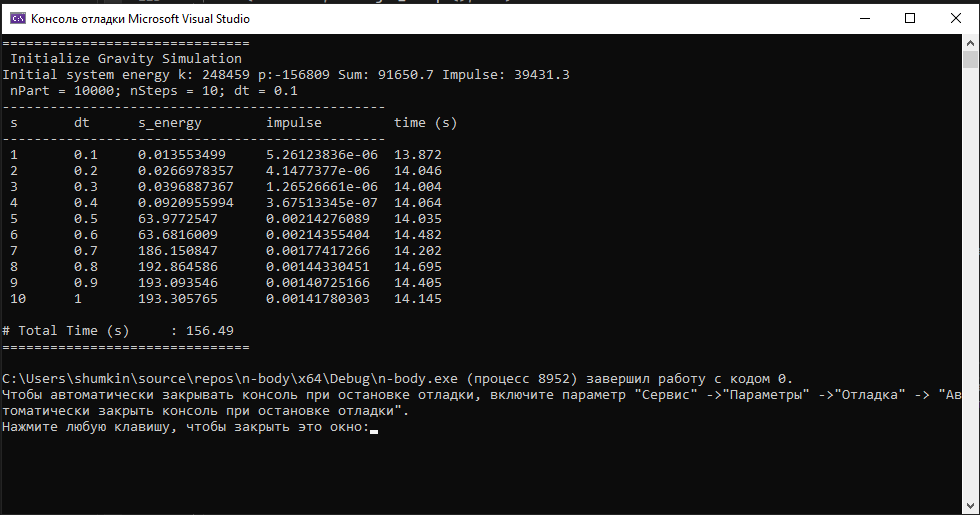


Figure 1 До изменений

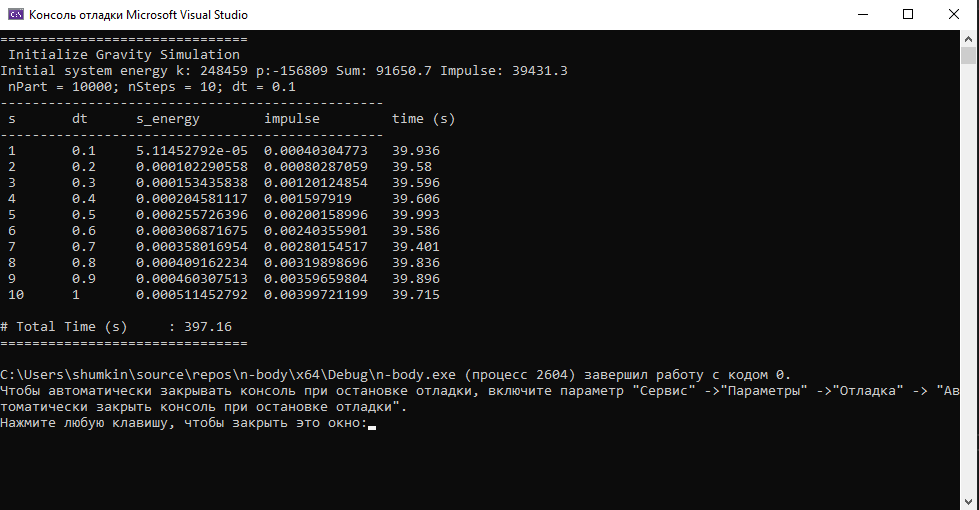
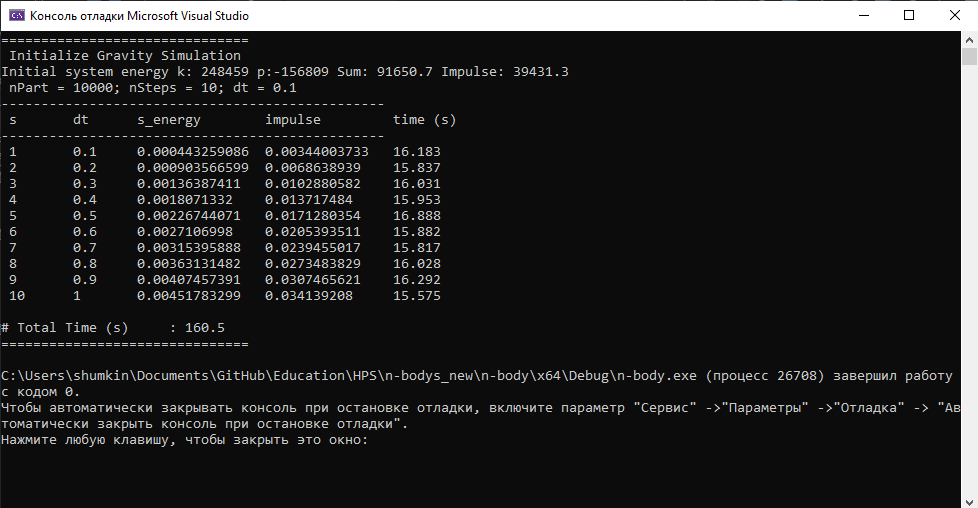


Figure 2 Реализовал метод Хойна



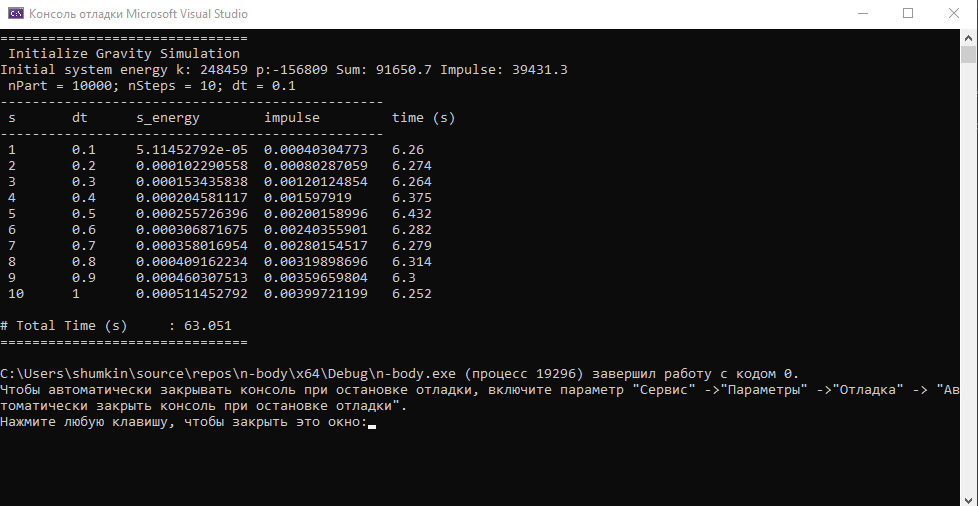


Figure 3 Заменил pow на умножение

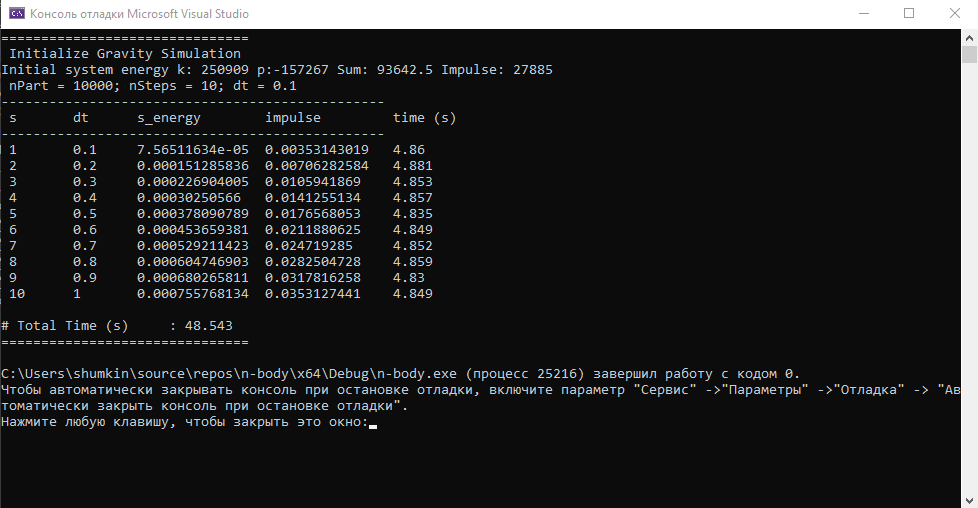
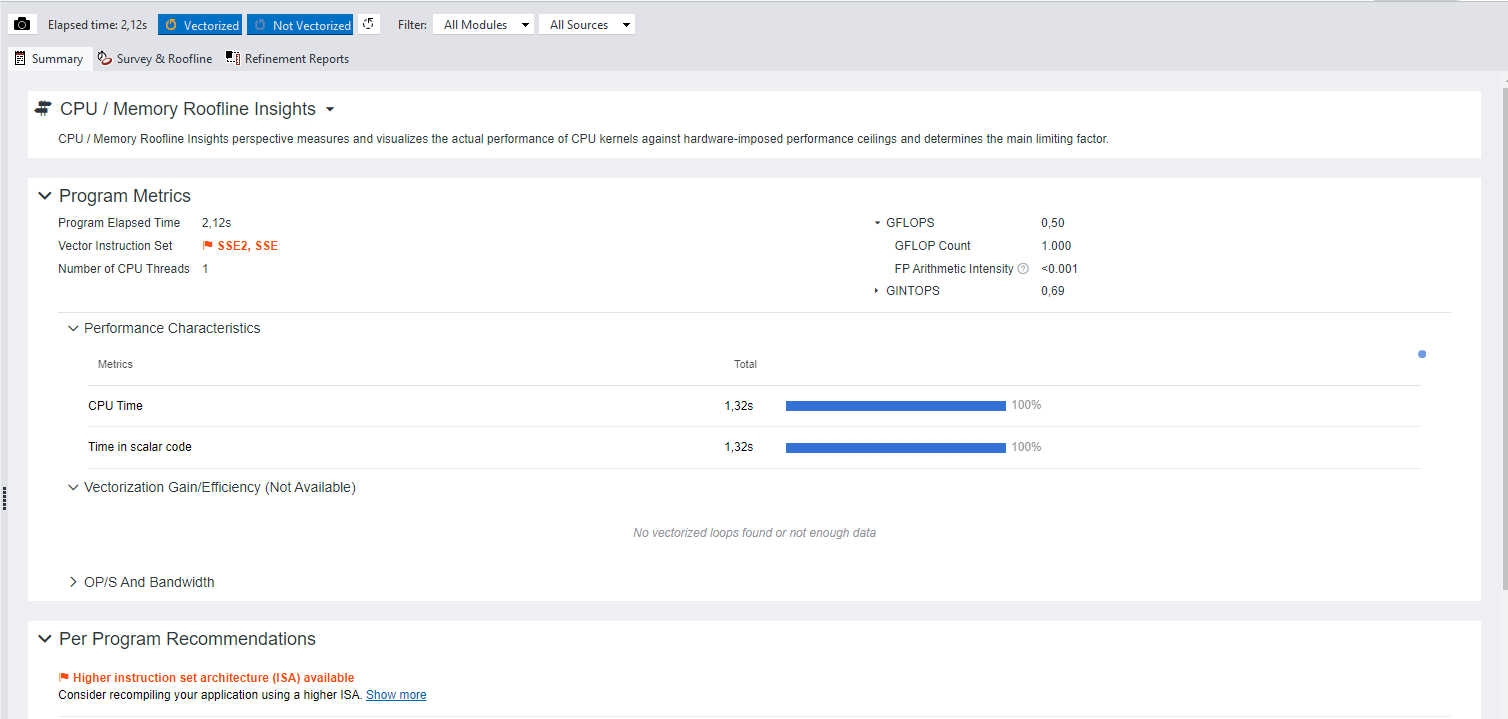
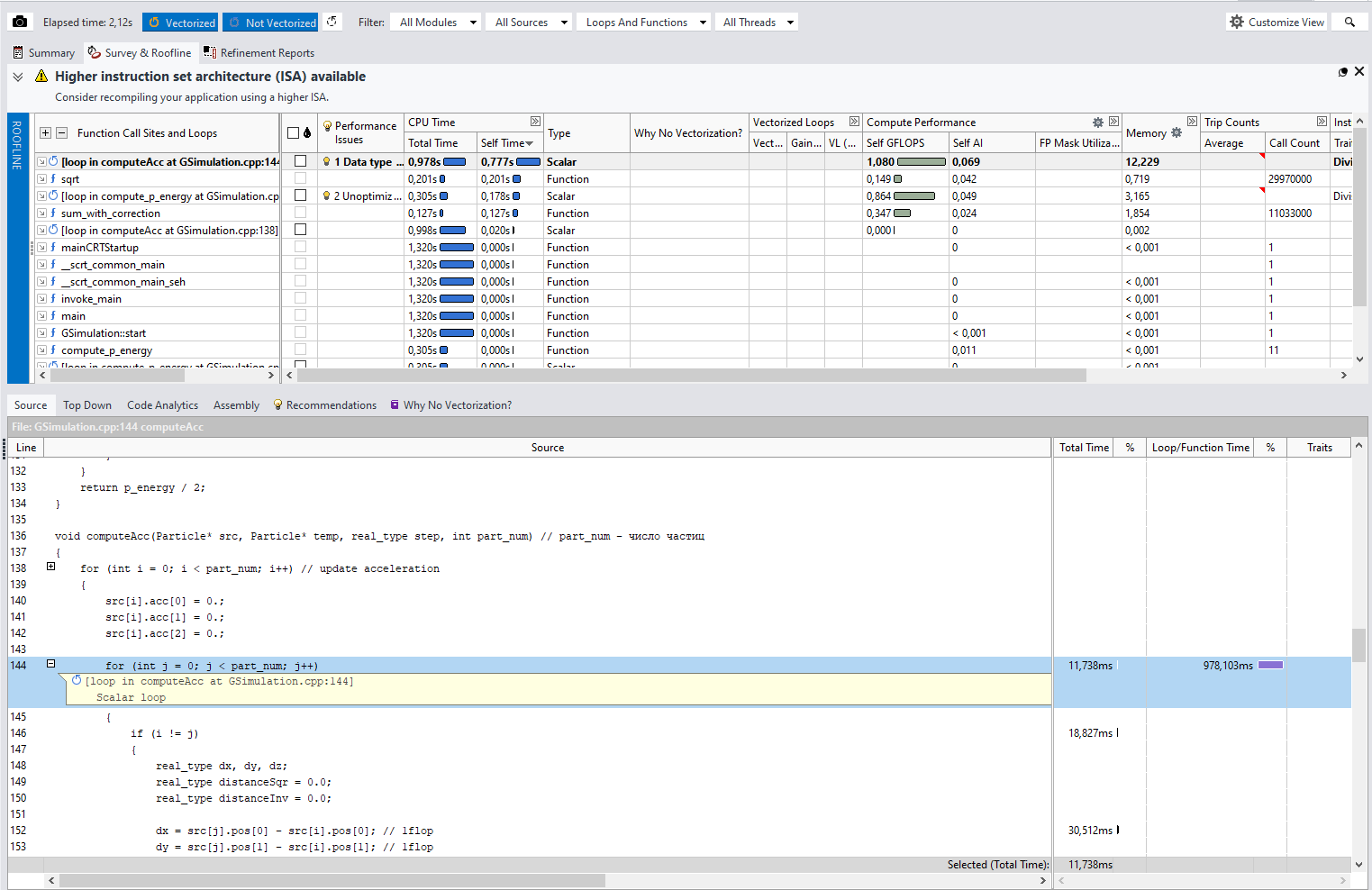
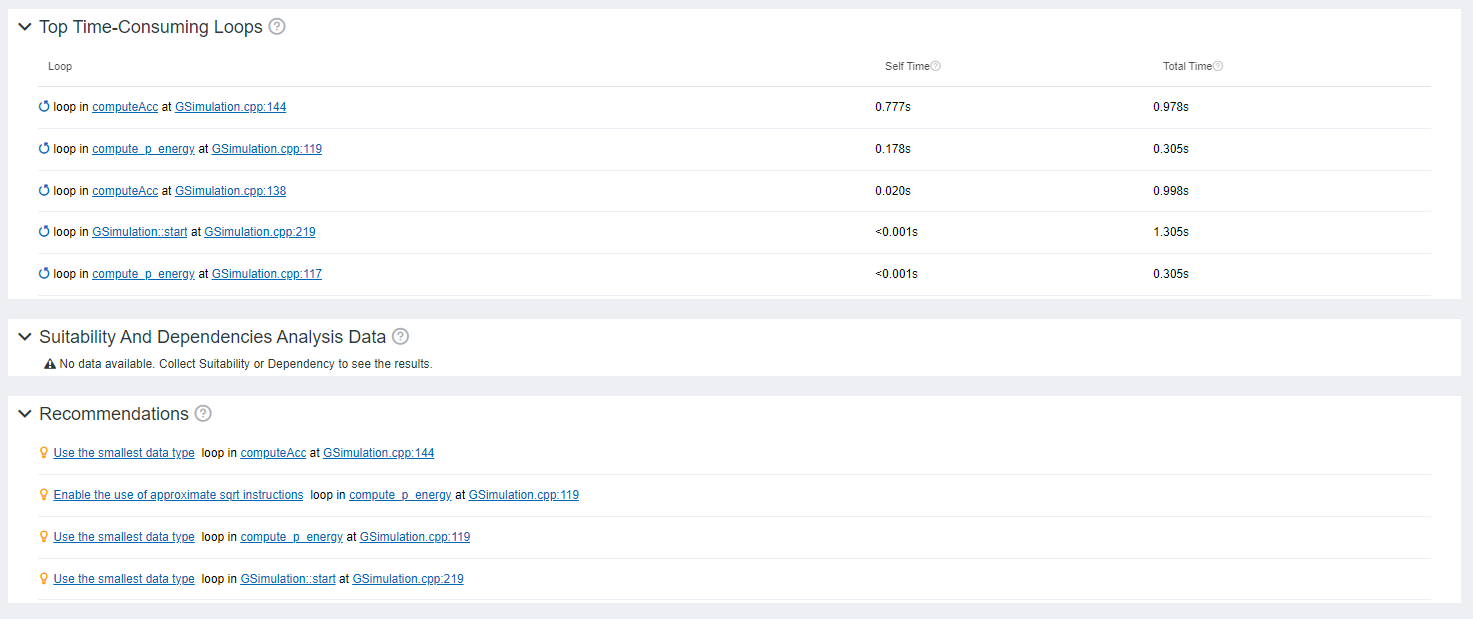
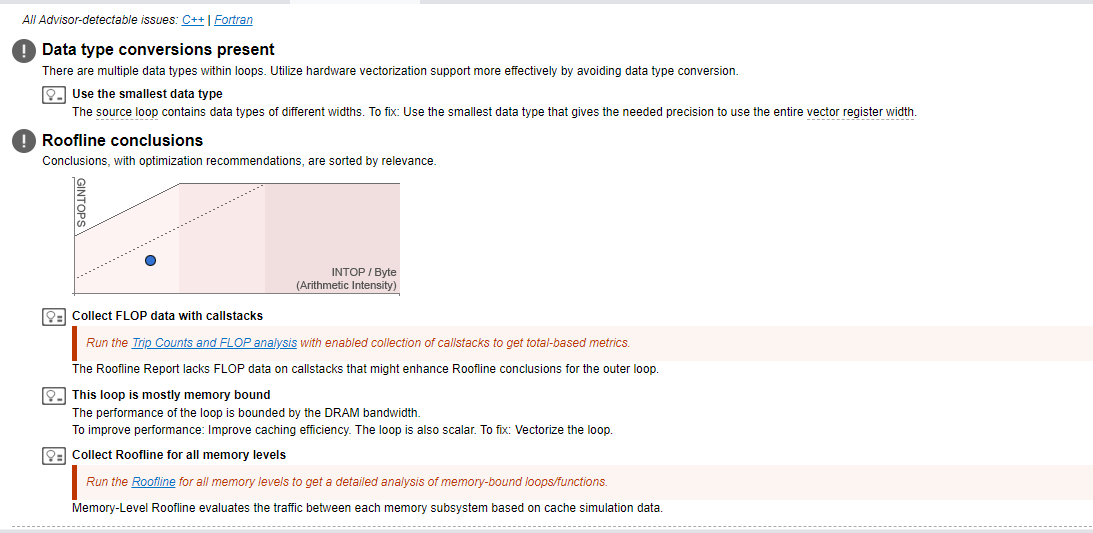
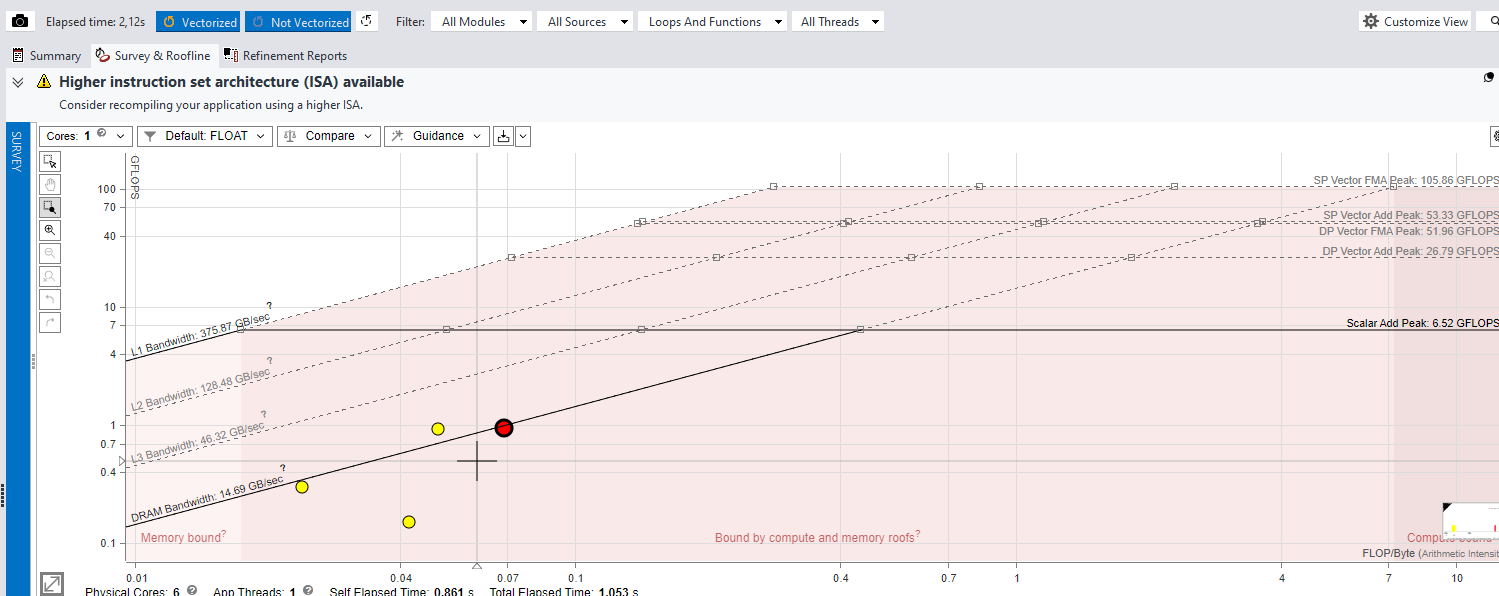


Figure 4 Заменил тип float на double

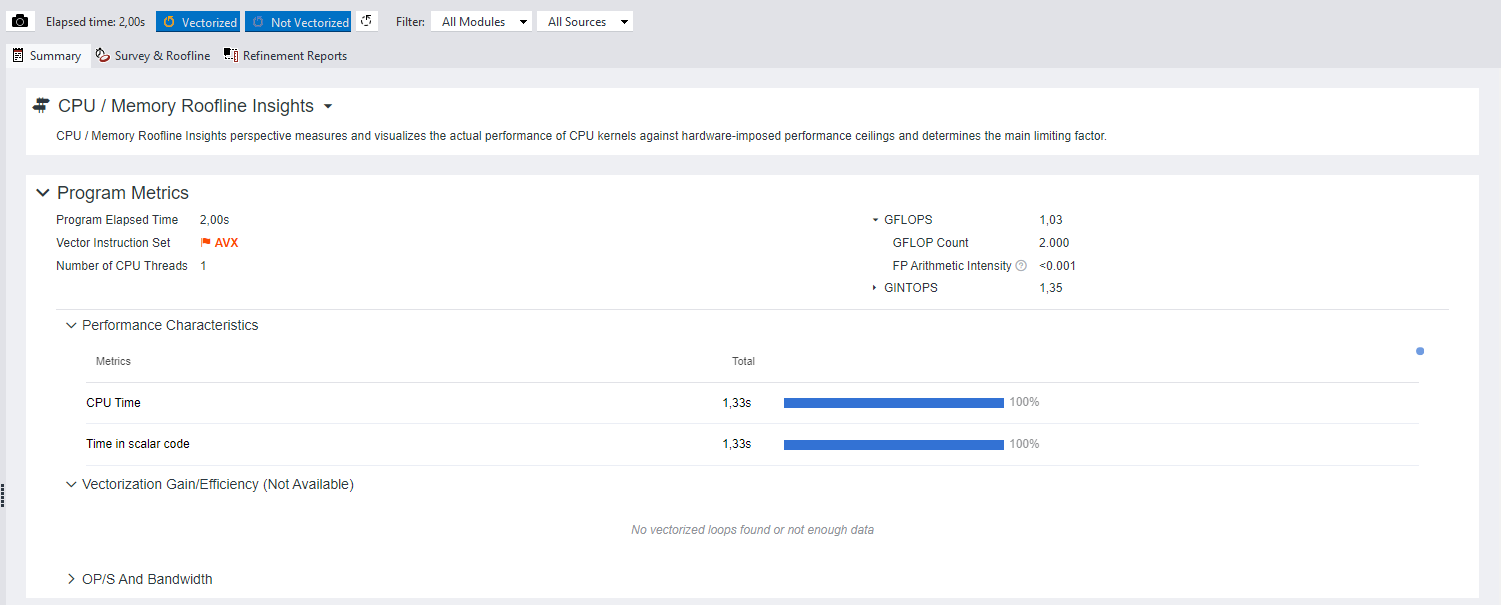




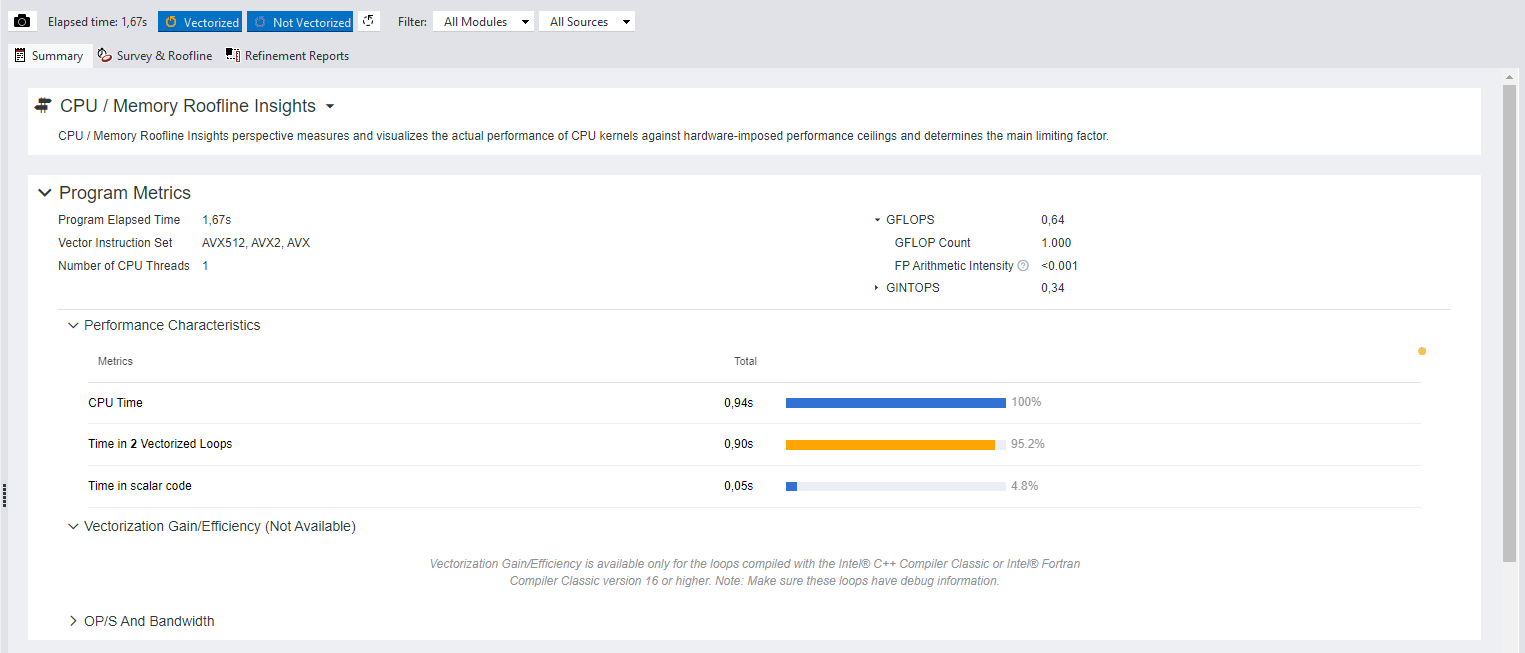


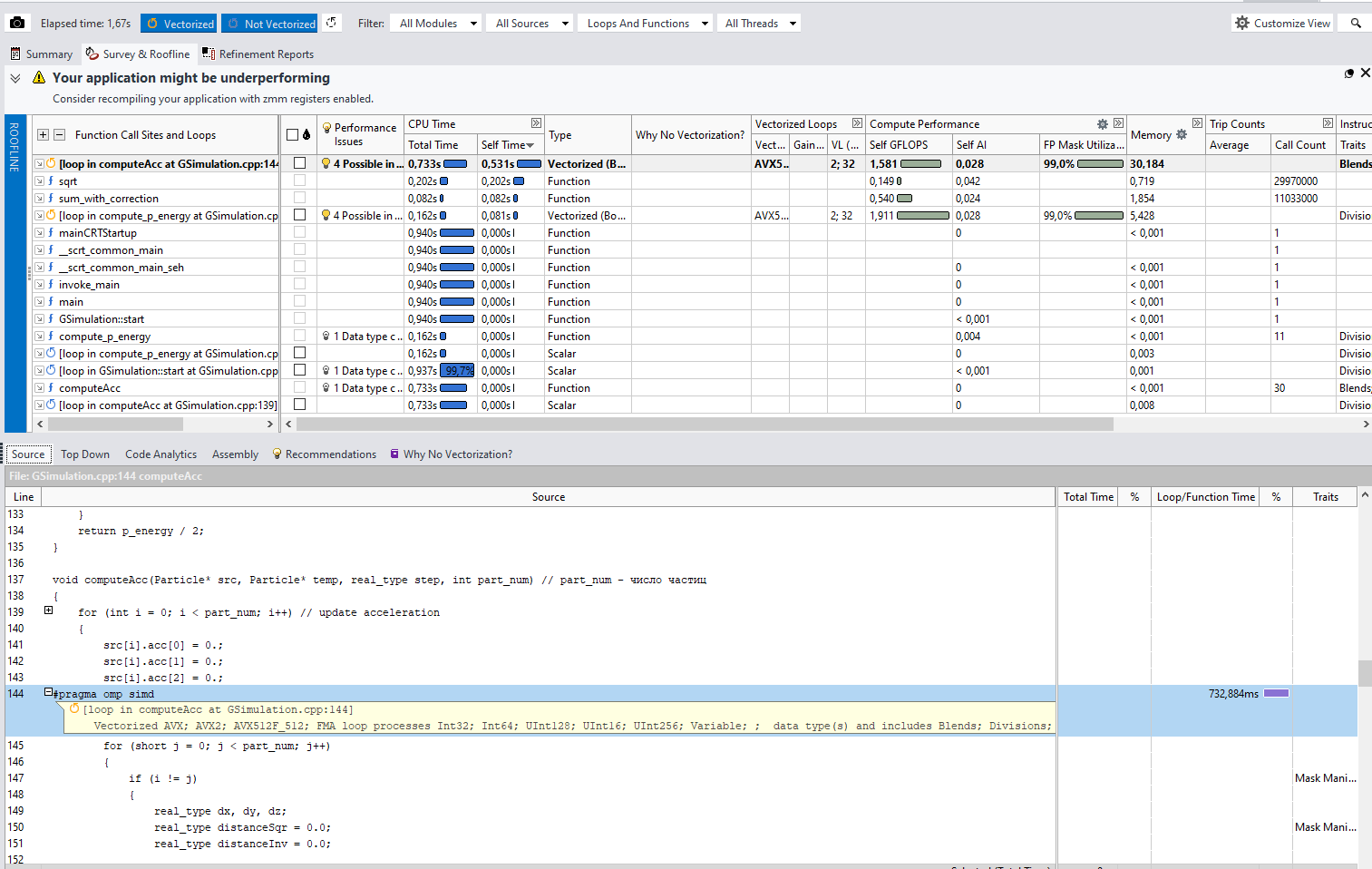


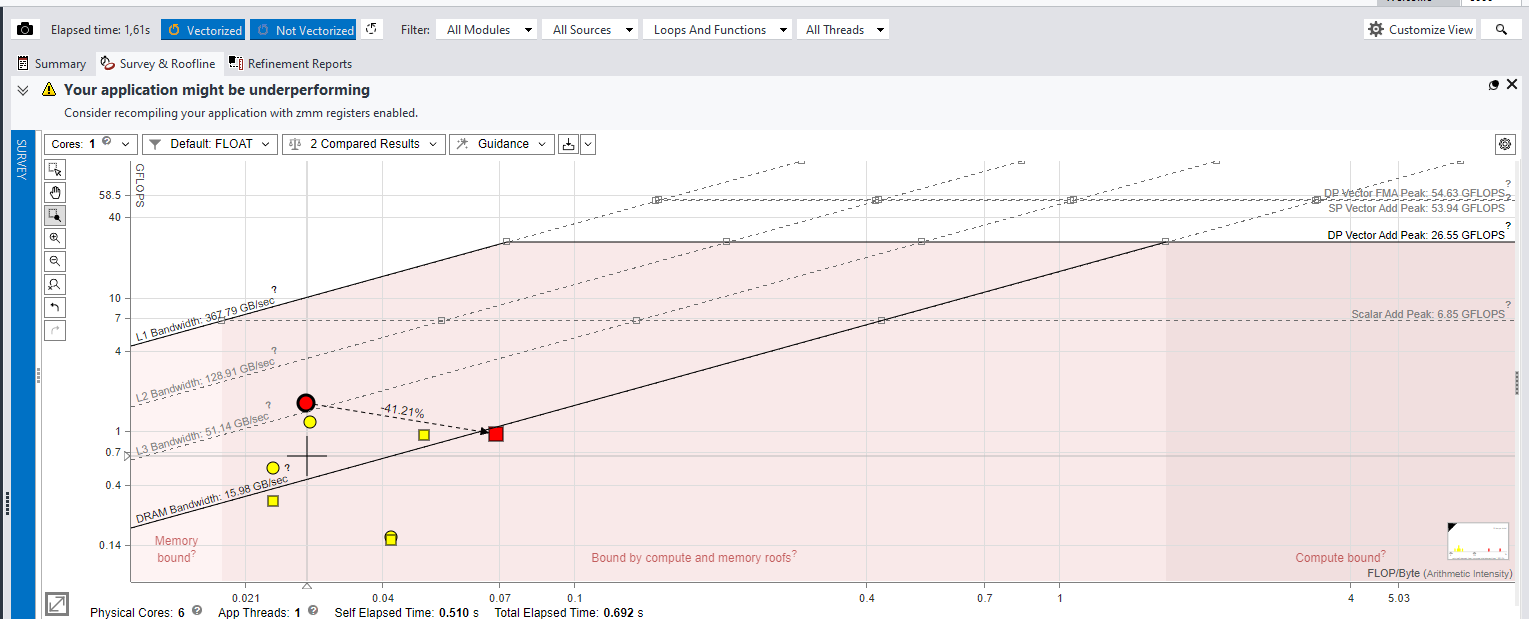
/QxCOMMON-AVX512

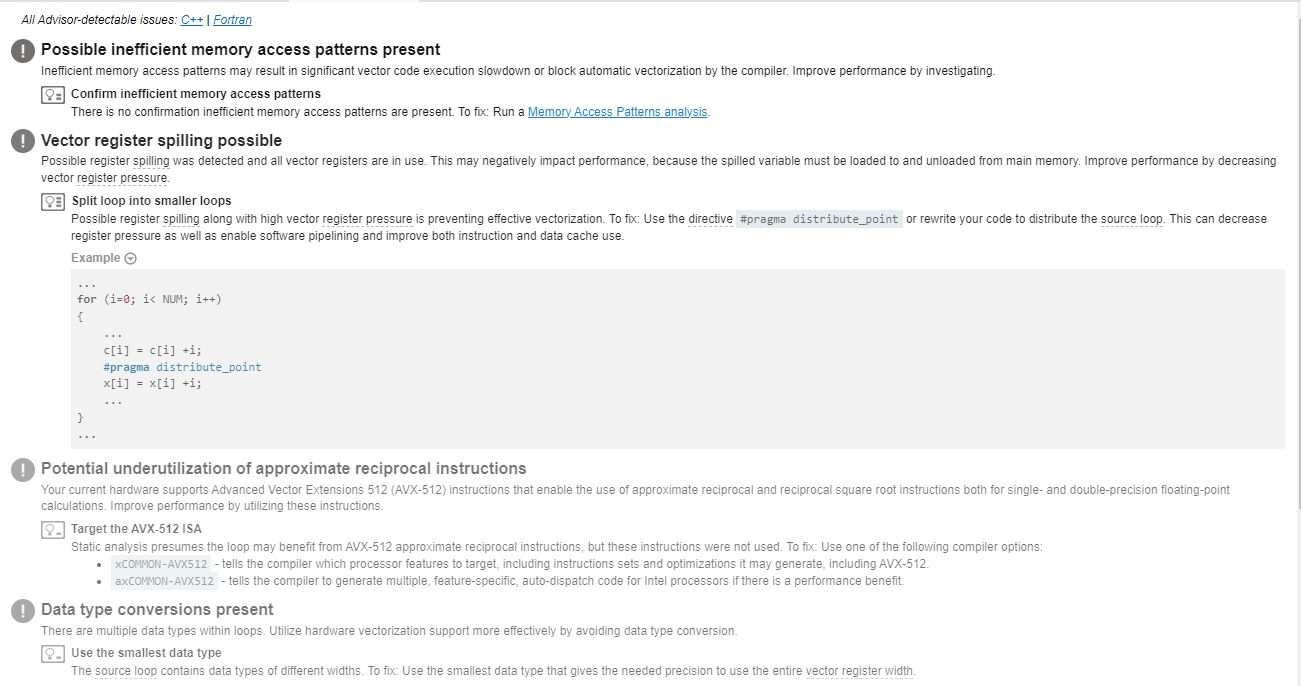


Добавил симды

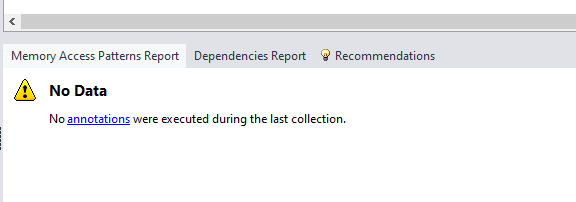




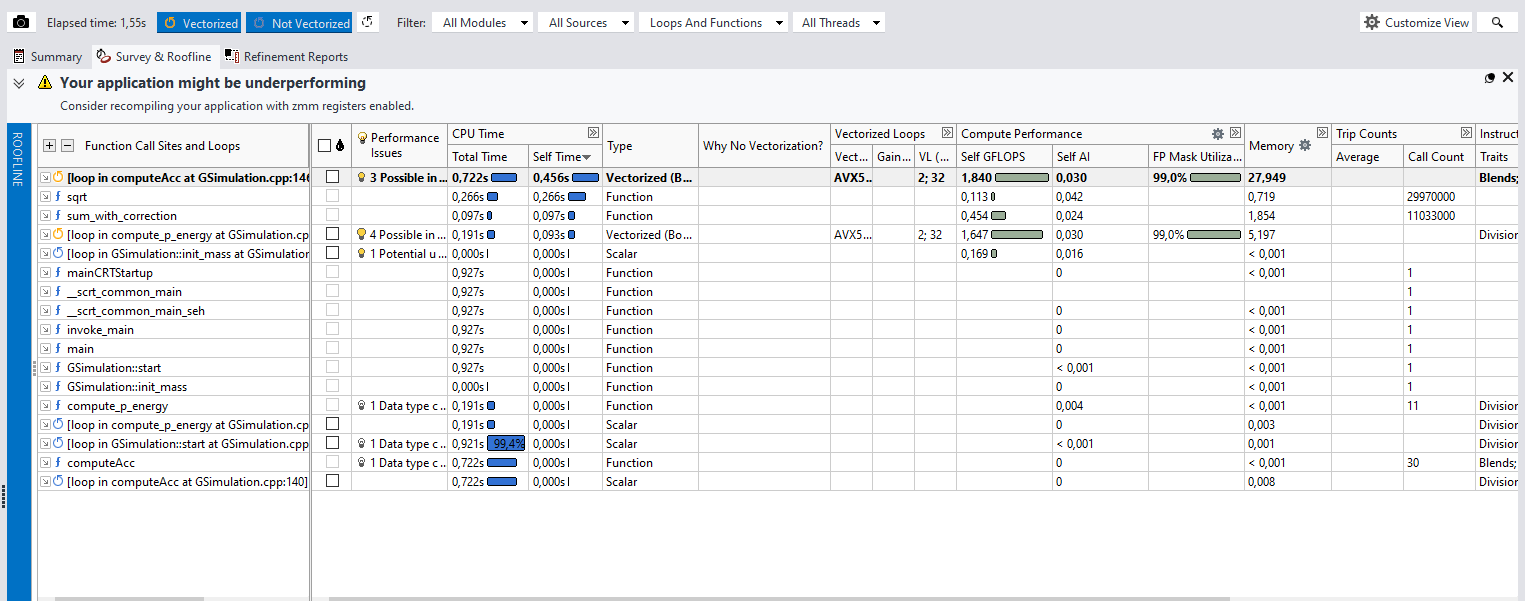


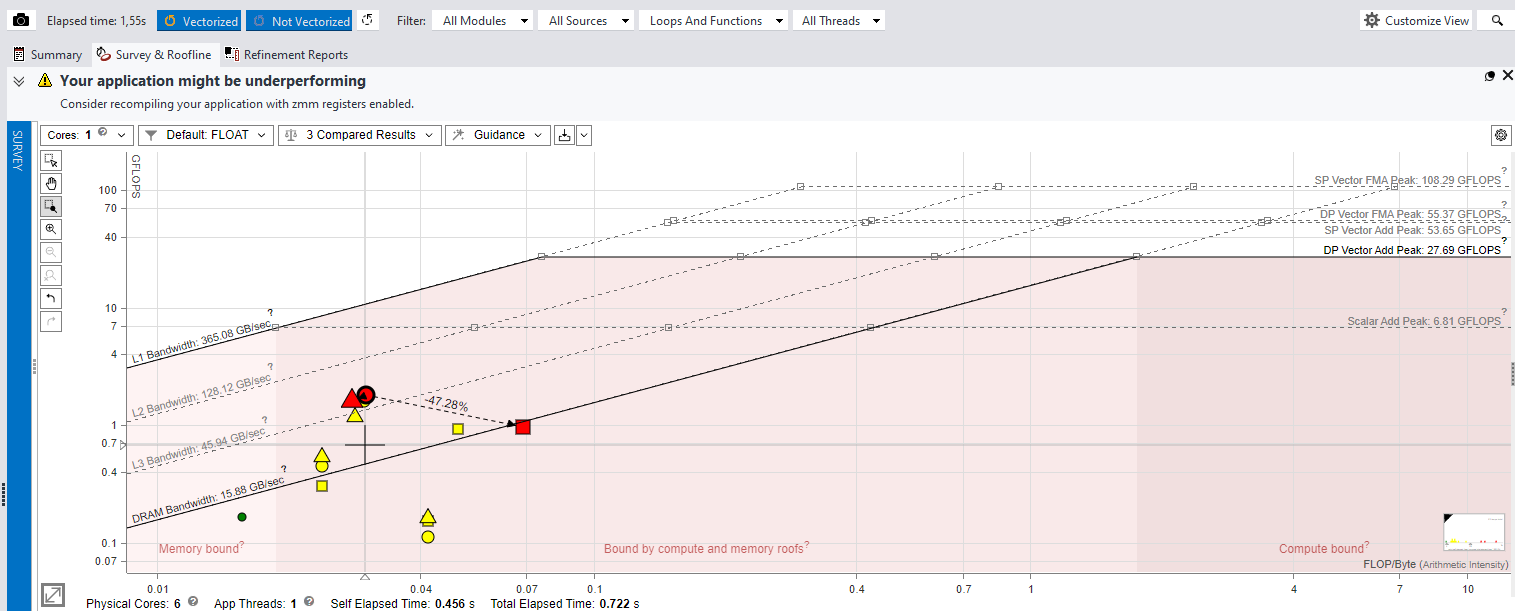


#pragma distribute\_point не сработал

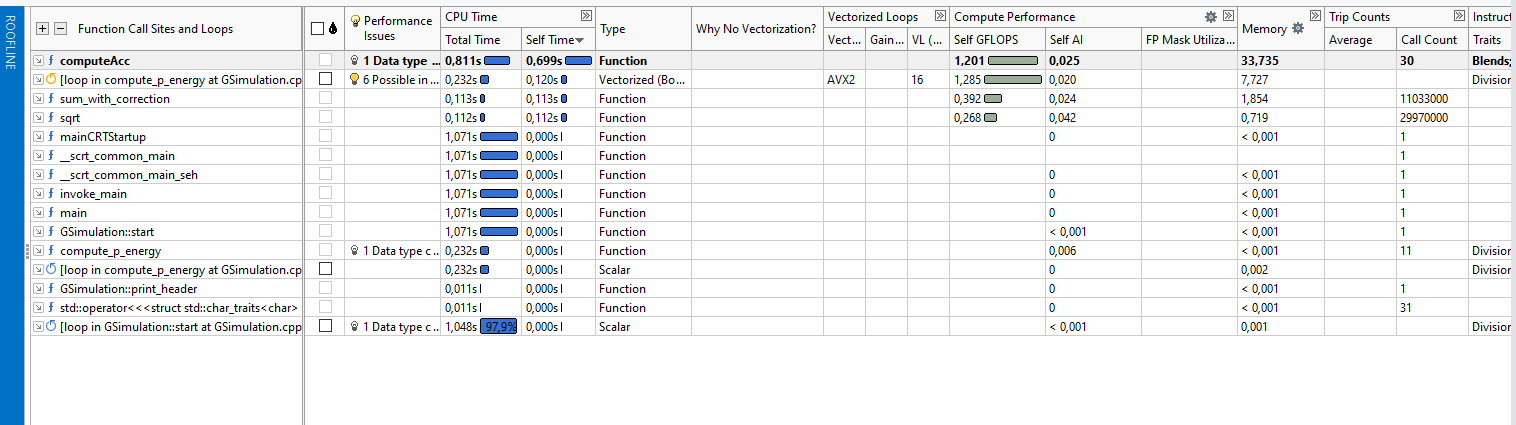


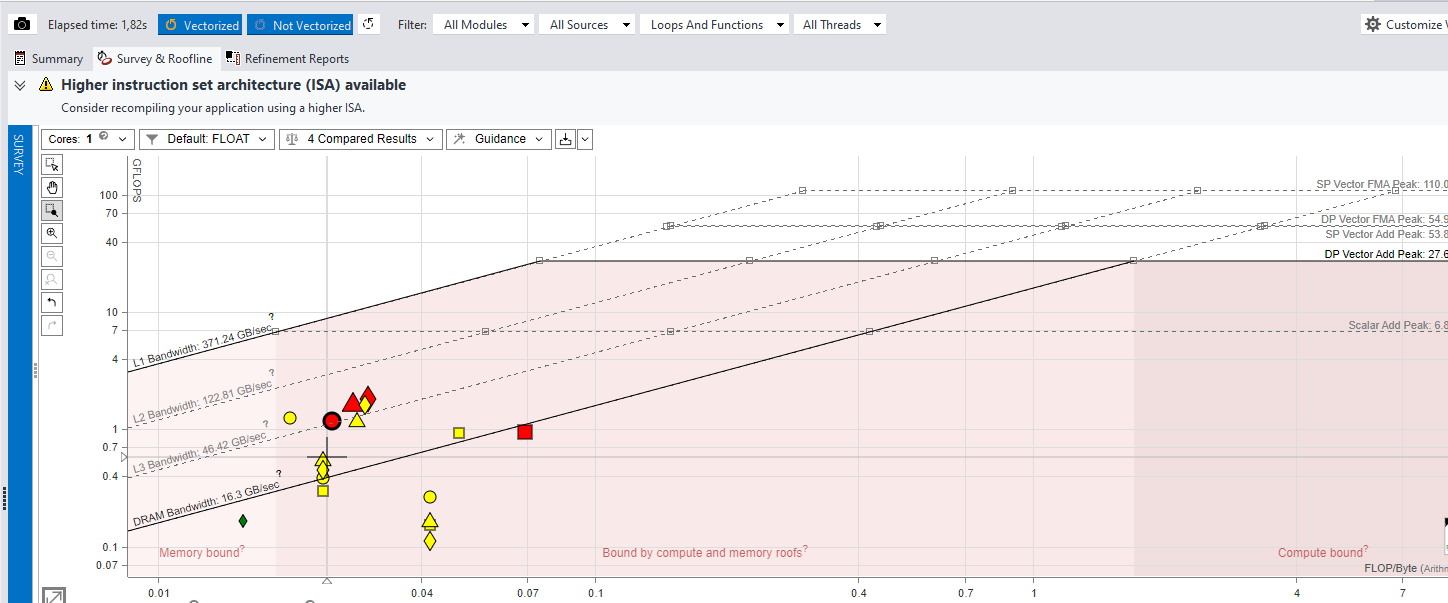
Уменьшил



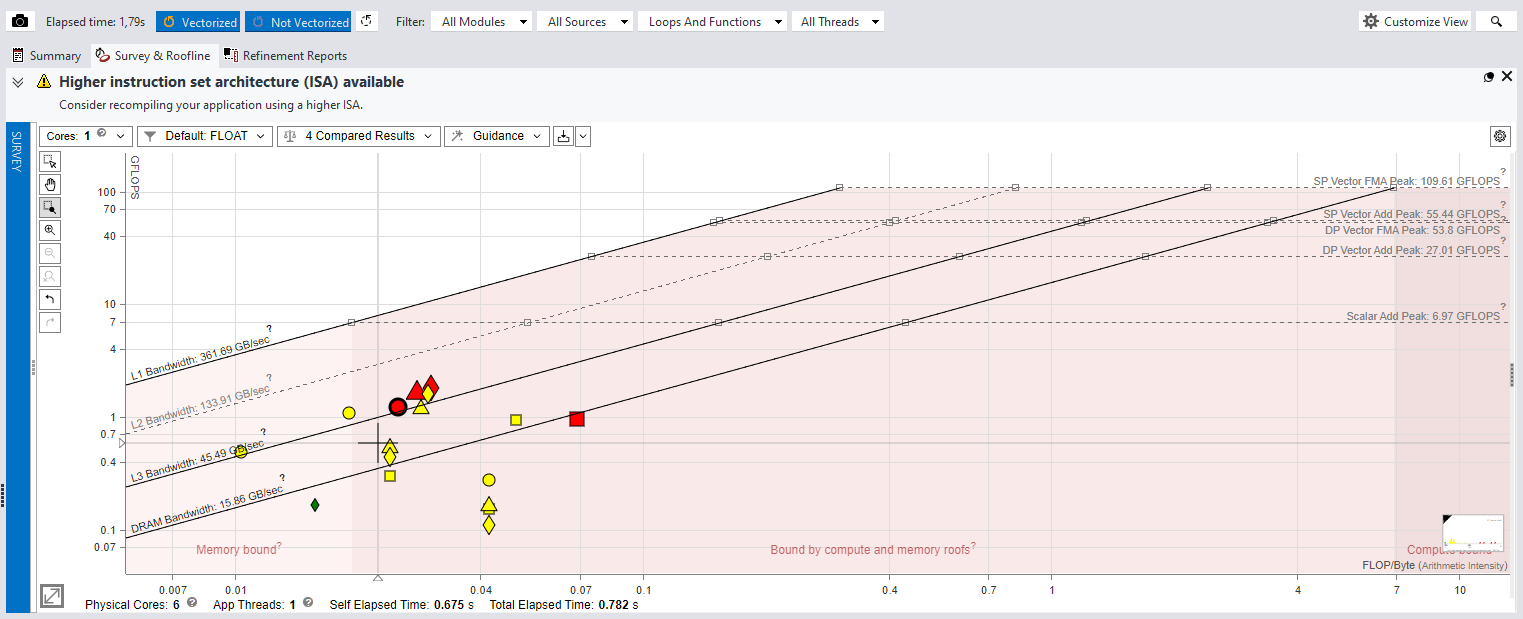
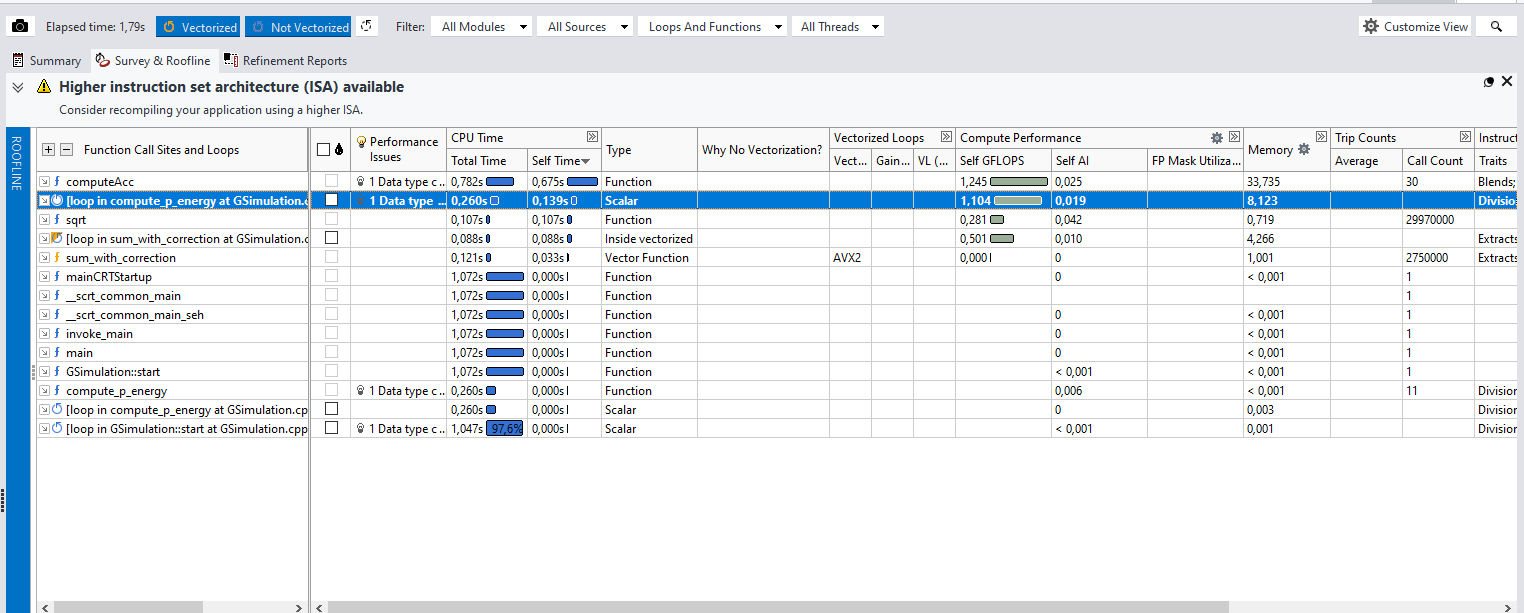


/QxCORE-AVX2 – стало хуже



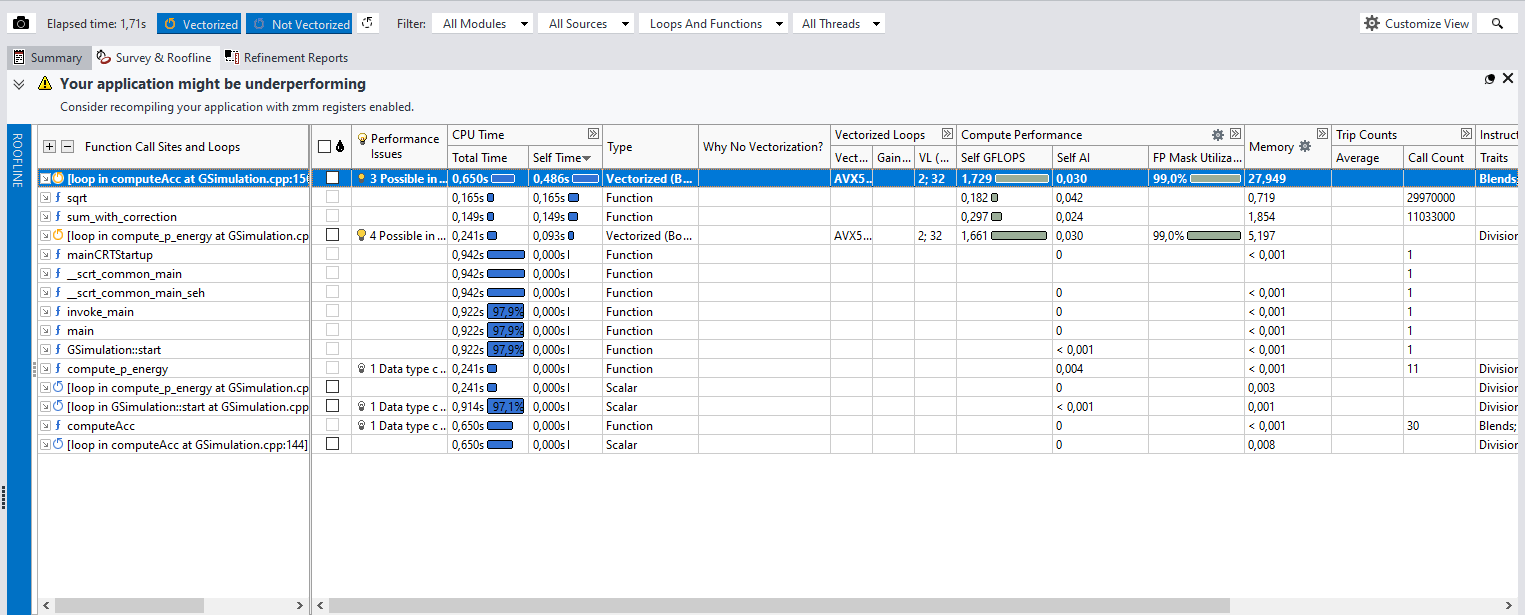


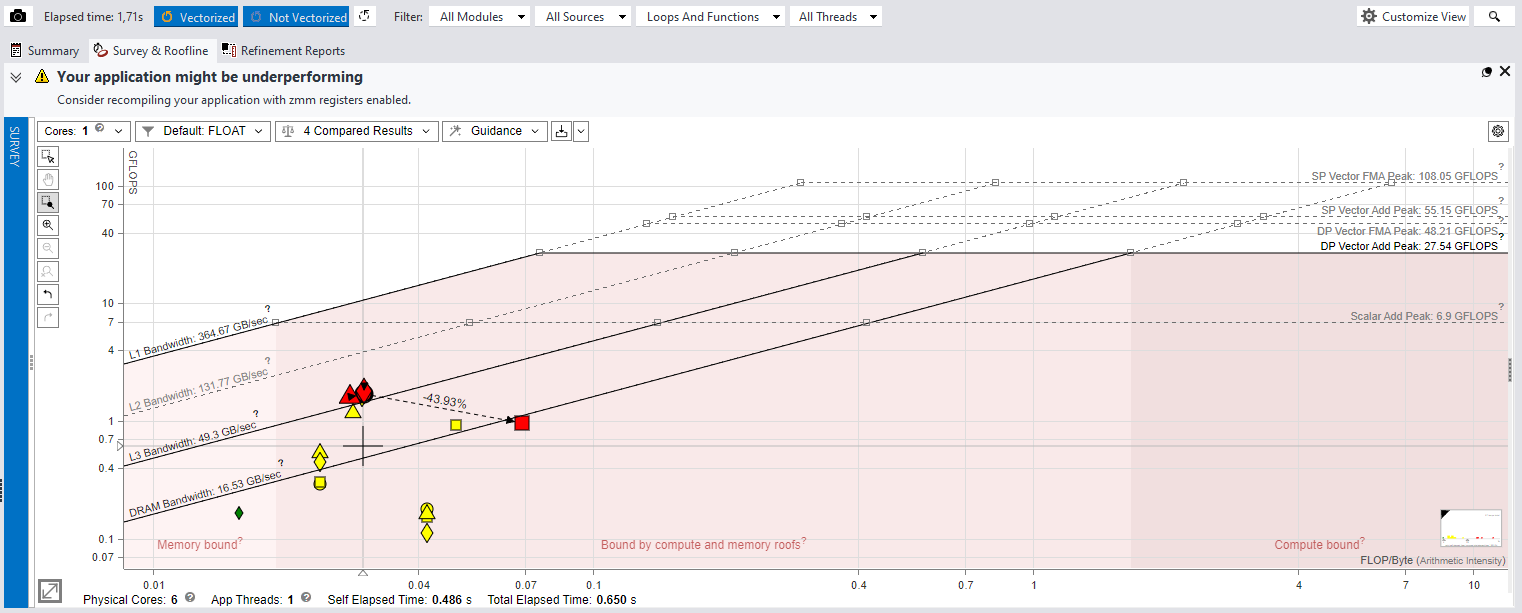
#pragma omp declare simd – для AVX2 на 10к стало получше



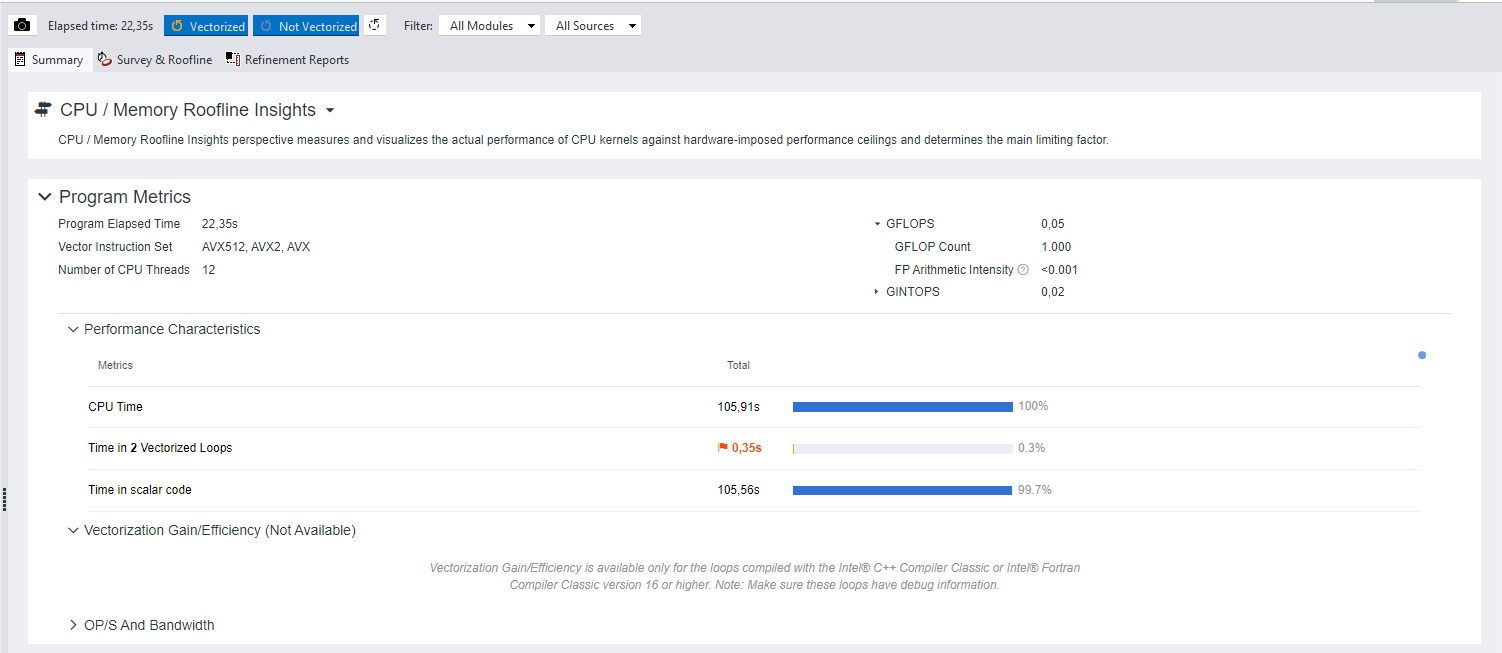
/Qprec-sqrt- - не запустилось

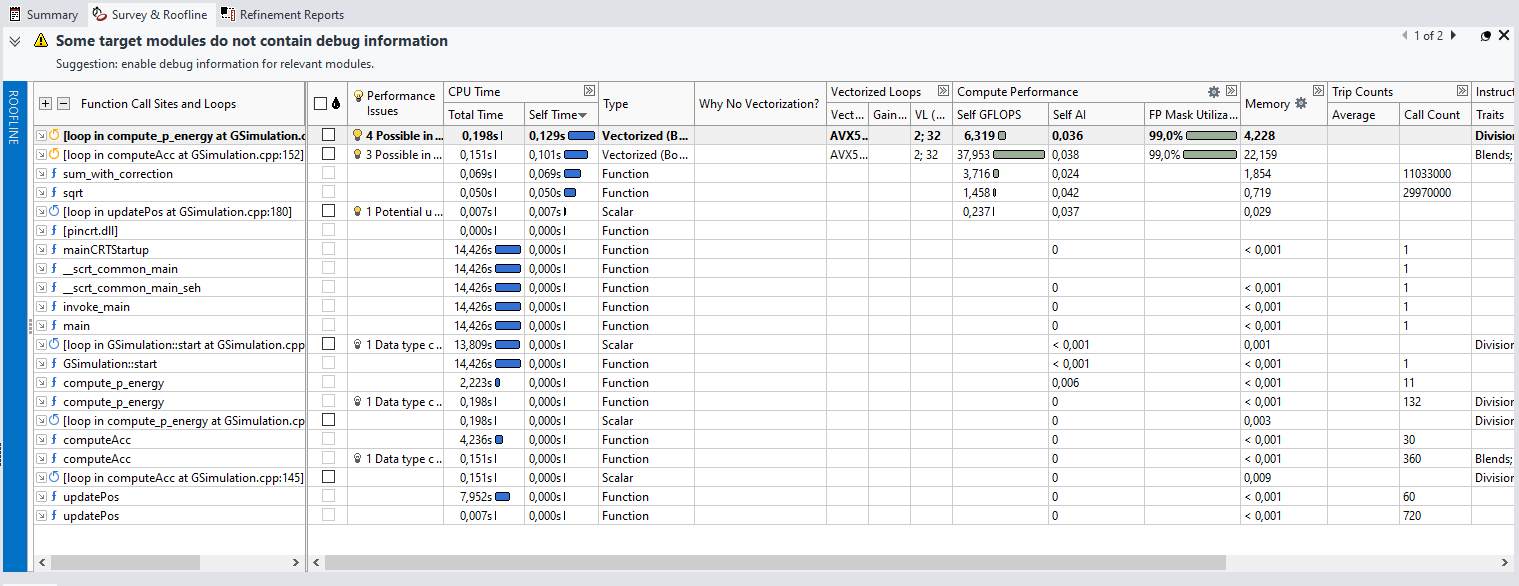
#pragma vector aligned #pragma vector always – не помогло

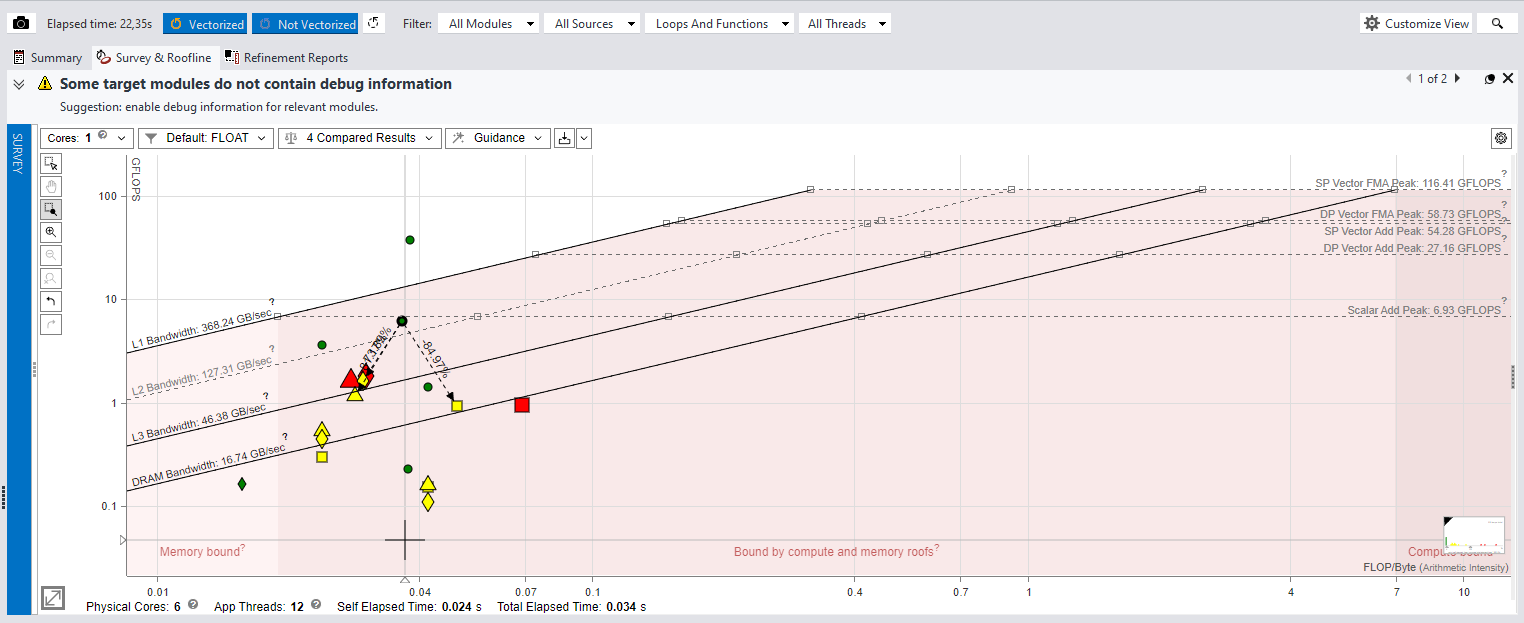




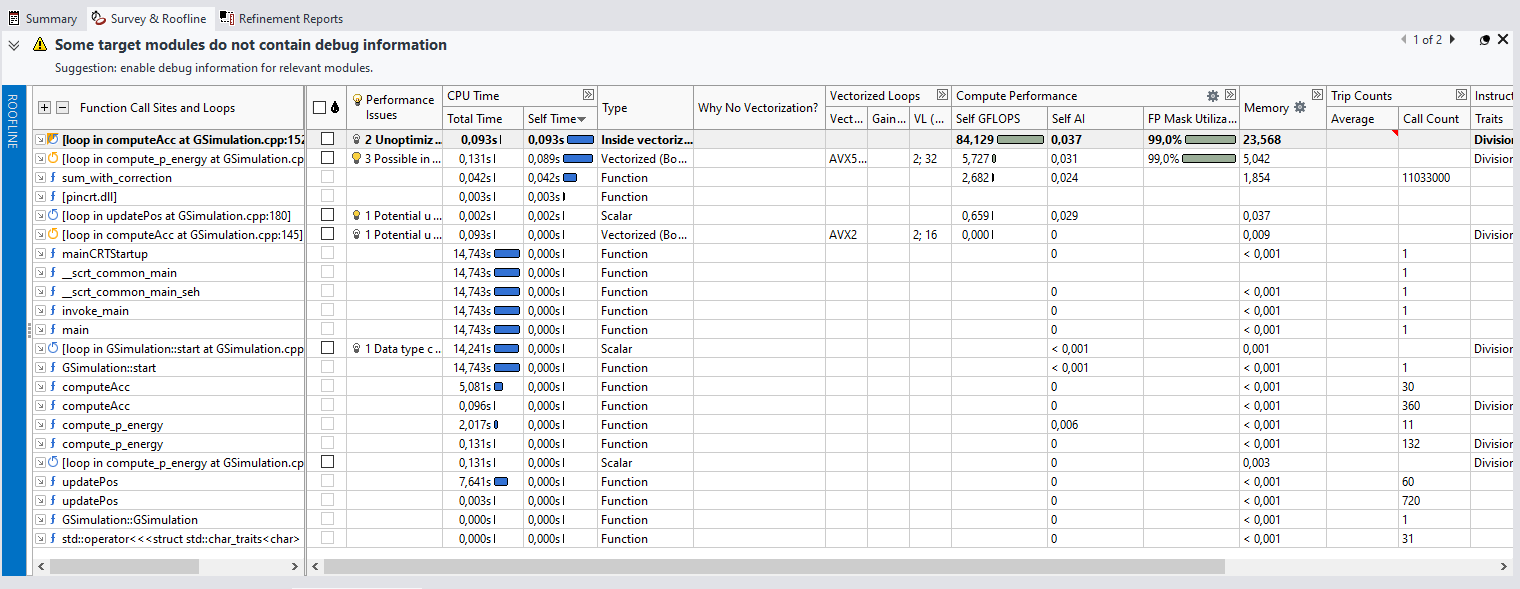
#pragma omp parallel for

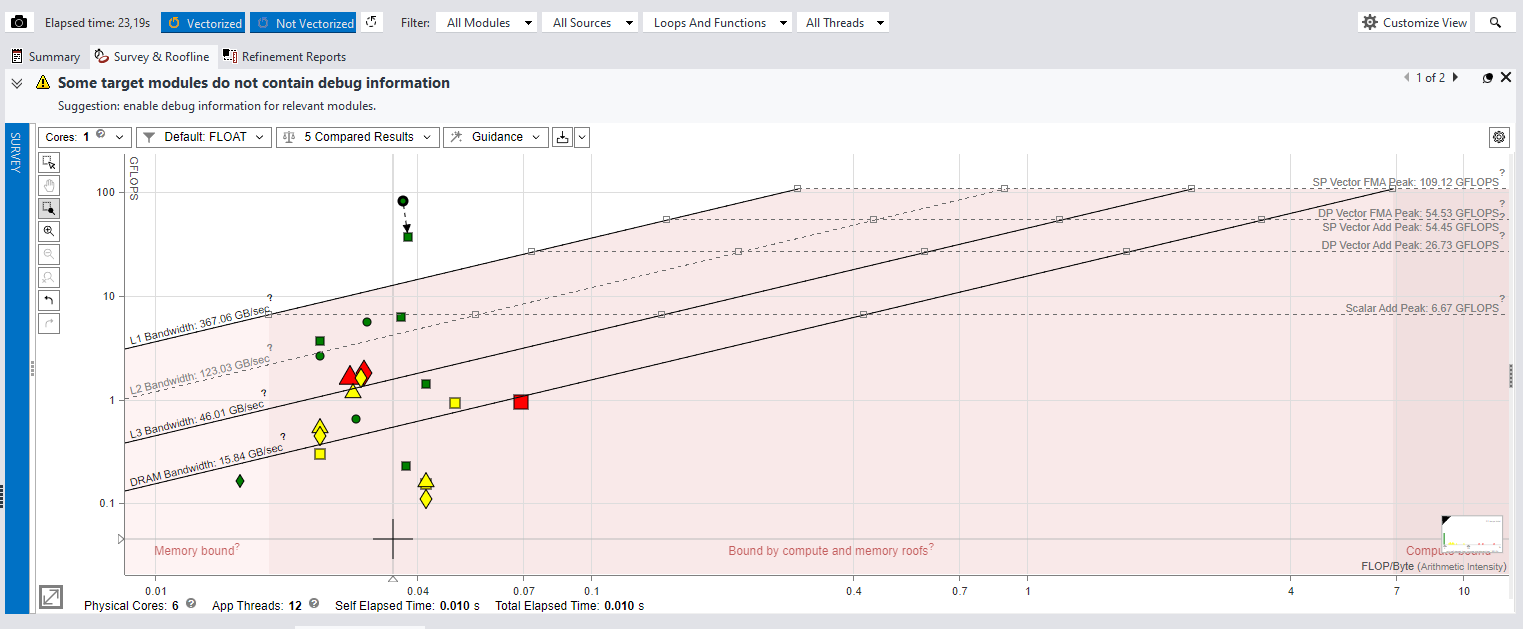




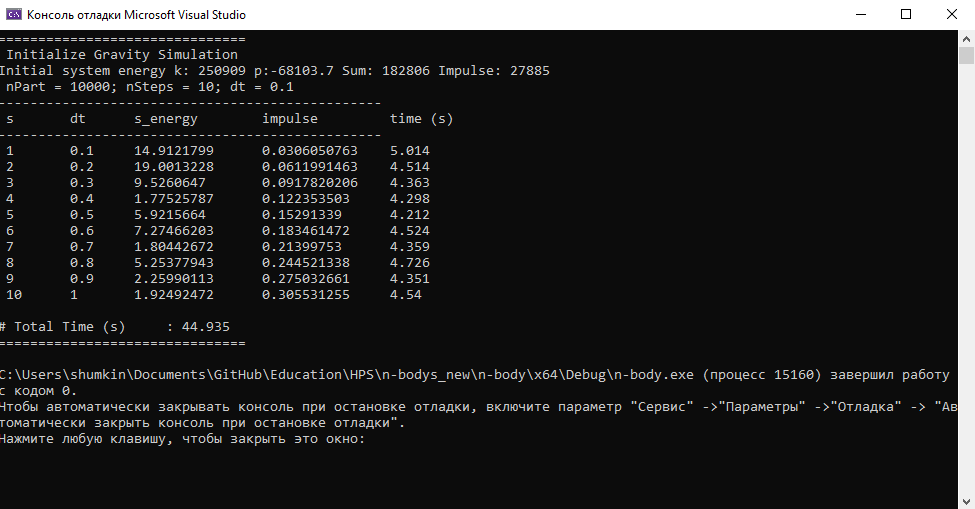


Заменил float на double в types.hpp

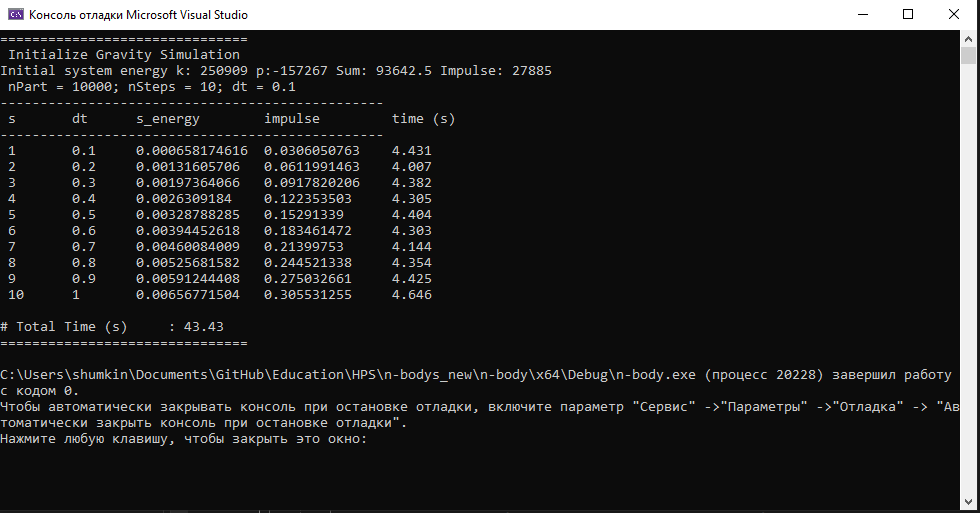


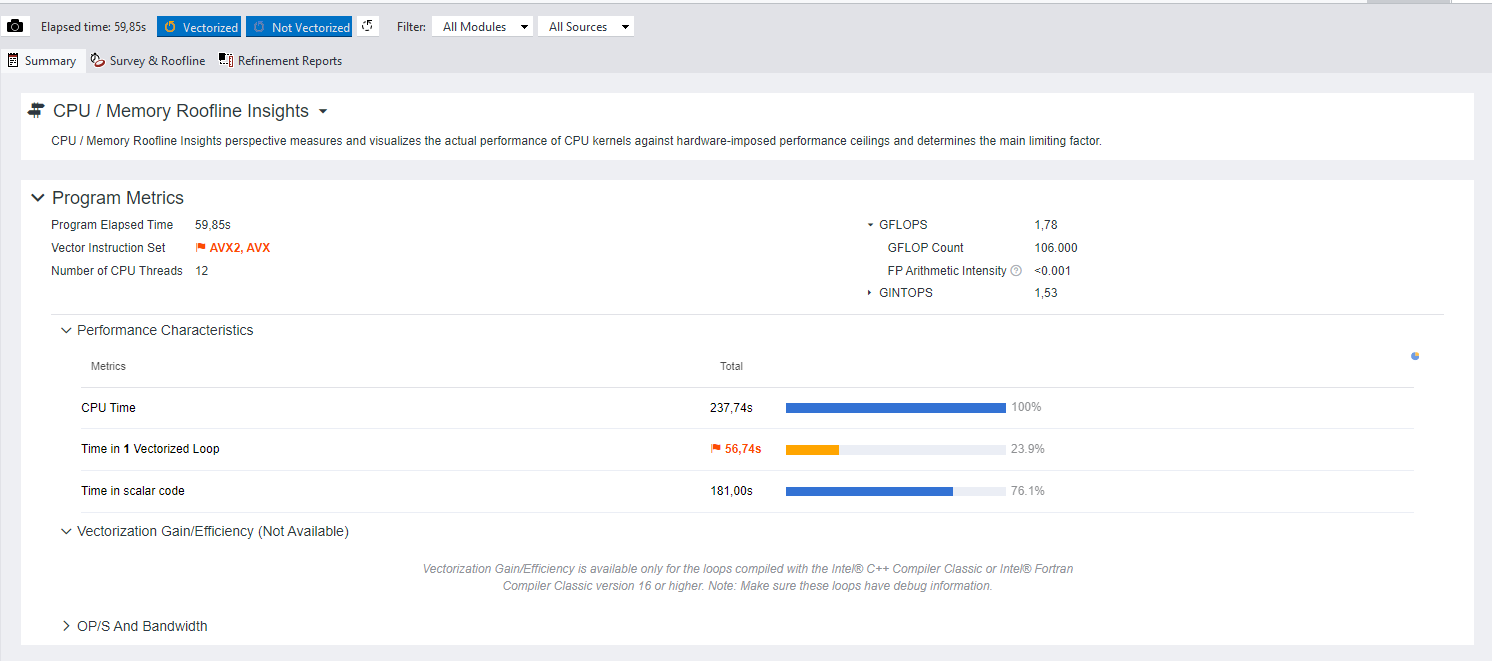


Итоговый результат:

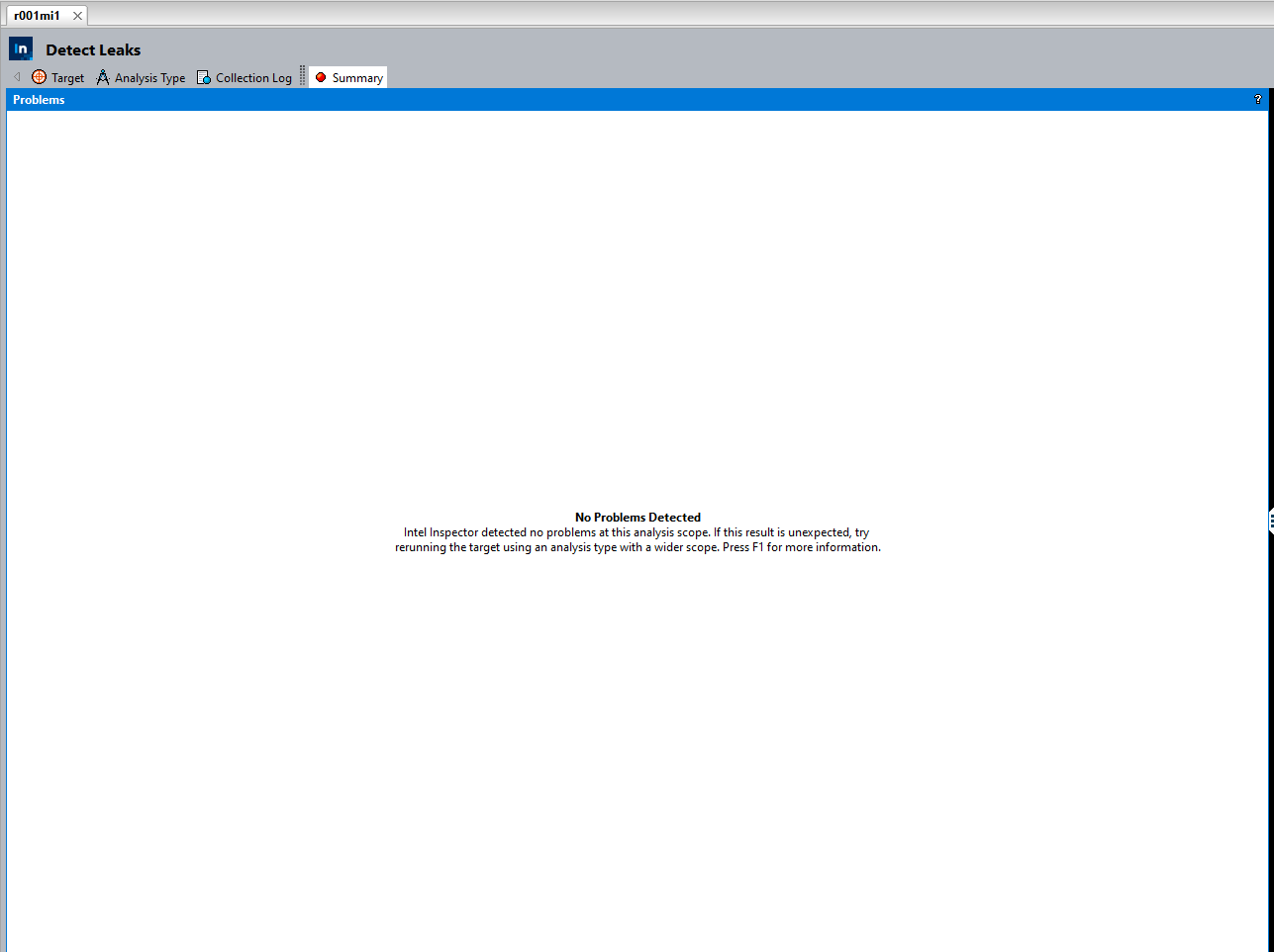


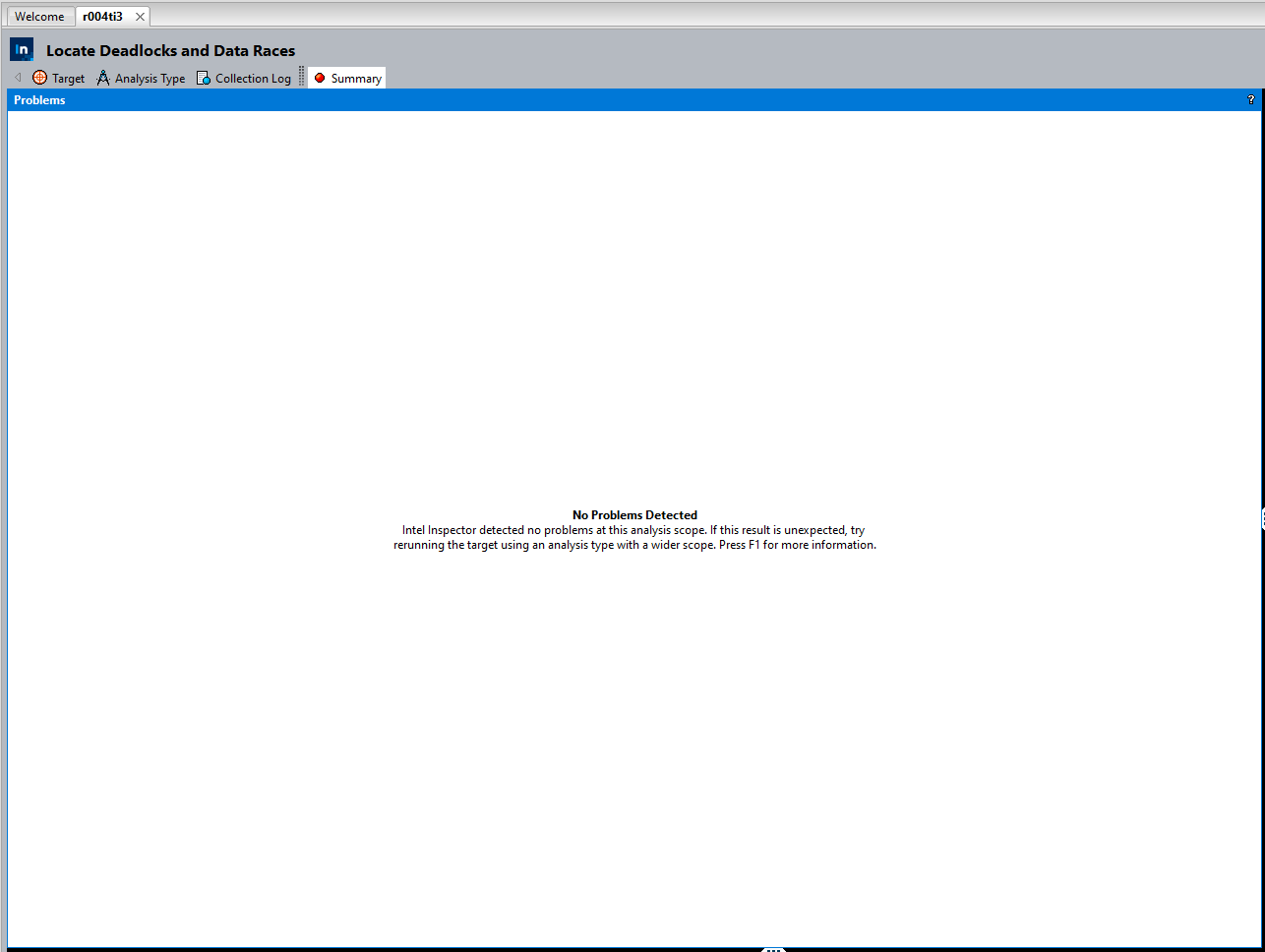
#pragma omp parallel for reduction(+ : p\_energy) помог избежать гонки переменных(нет)

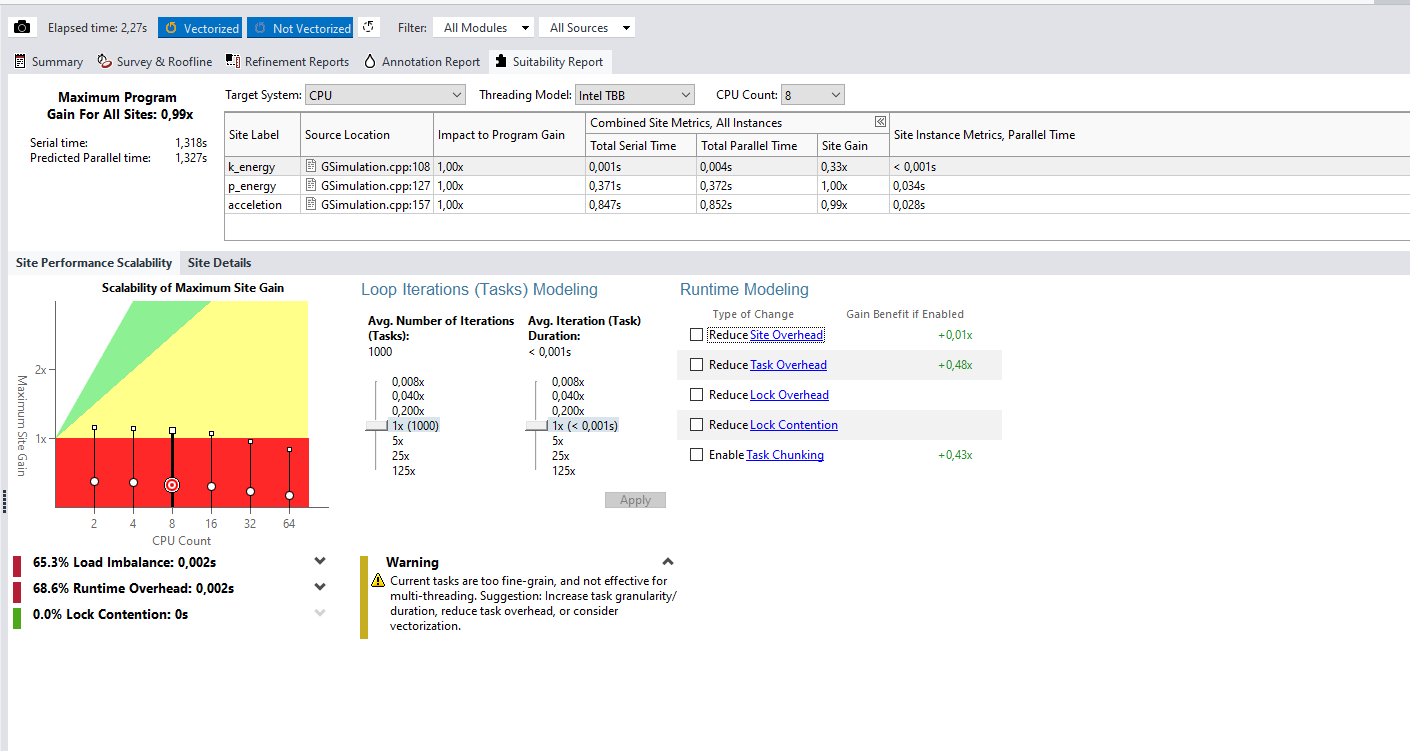


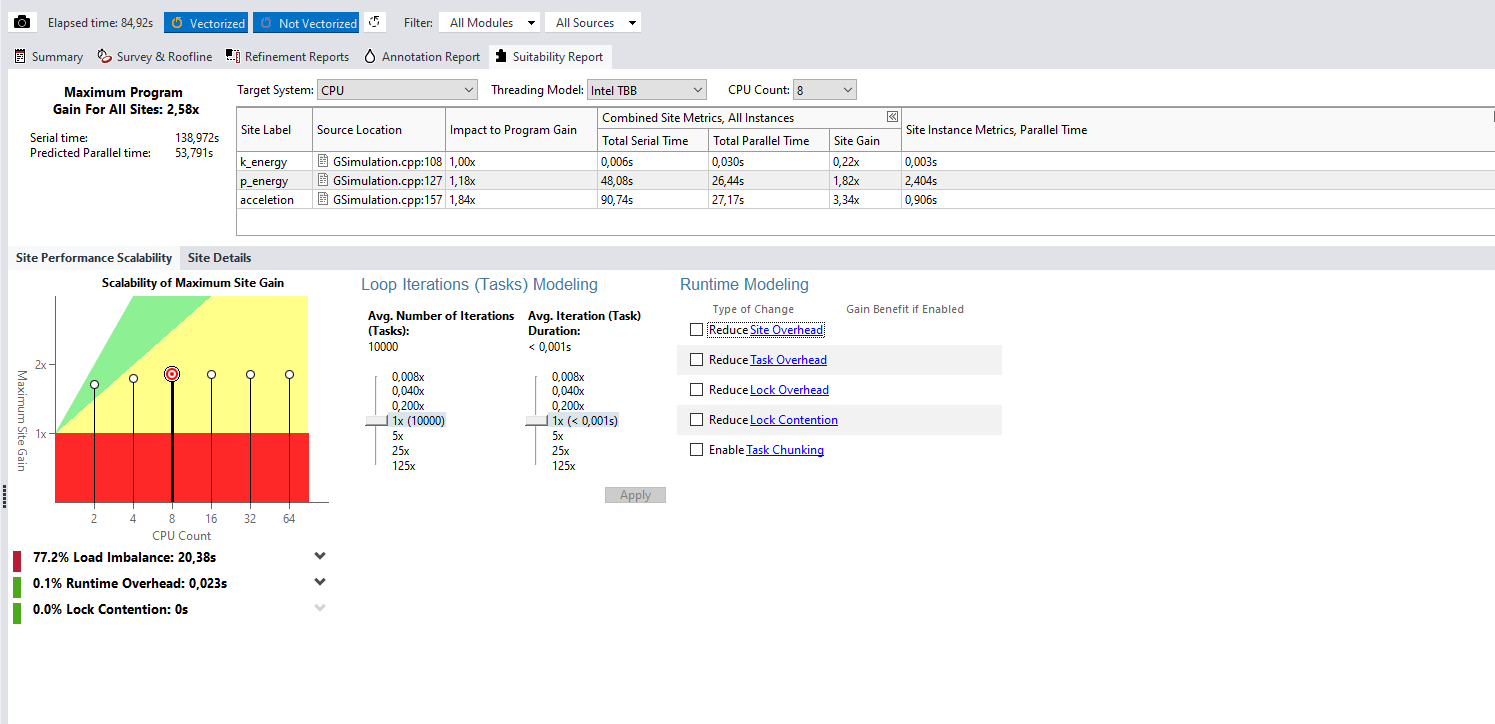
Для AVX2 при 10к

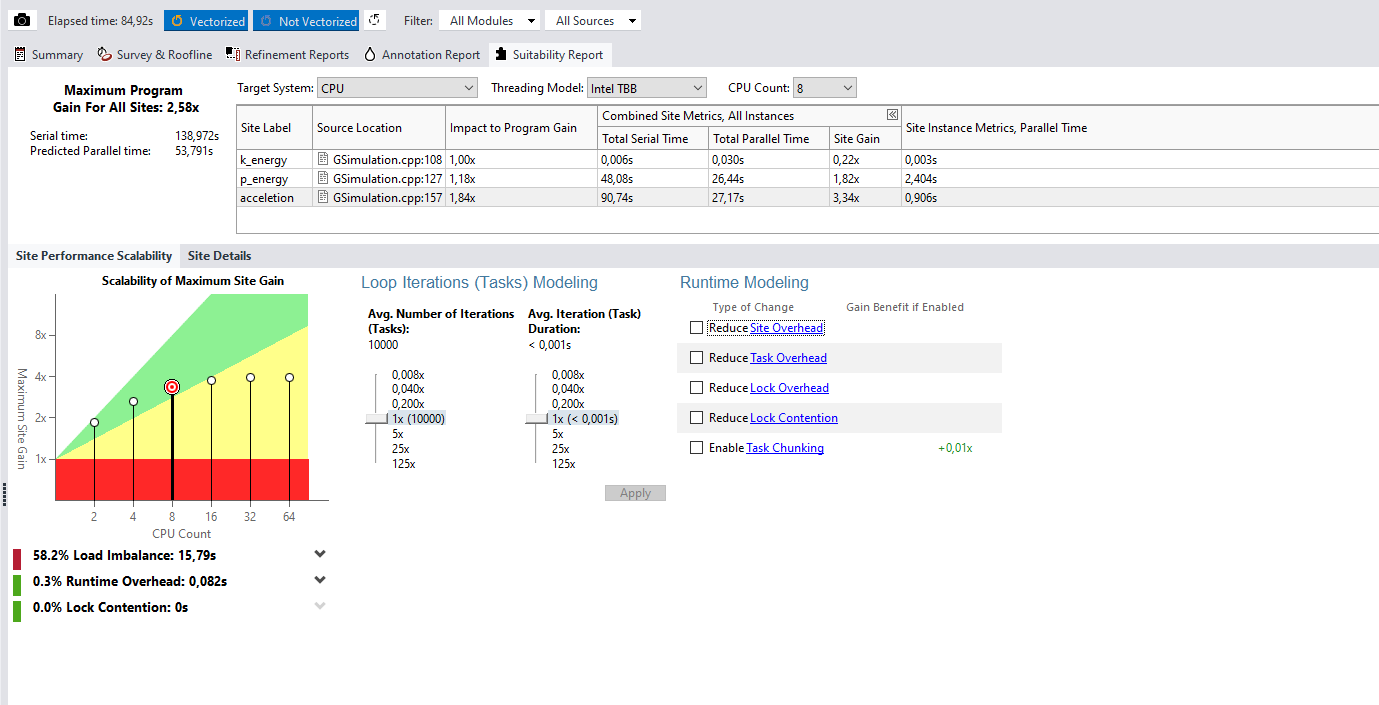


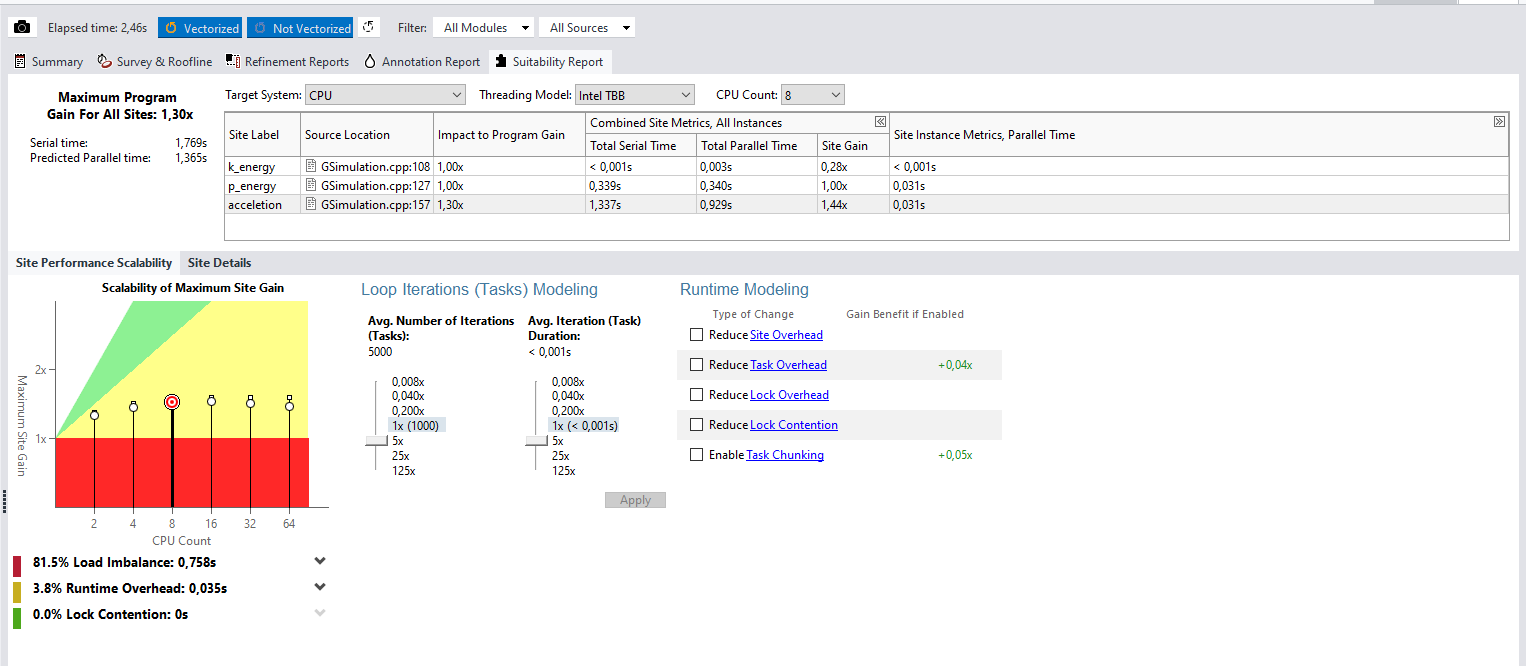












Вывод: при числе итераций ~10к параллелизм даёт прирост