Напишите параллельную программу вычисления следующего интеграла с использованием дополнений ***Intel Cilk Plus*** языка C++:

Integral_6.png

1. Описание проблемы и краткая характеристика инструментов параллелизации, используемых для решения задачи

Необходимо численно решить определенный интеграл , используя несколько потоков процессора . Для этого лучше использовать параллельный язык программирования (в данном случае дополнение для C++ - Cilk Plus ) . Во время написания программы нужно проверять код на различные ошибки , которые возникают в параллельной программе , а также провести оценку эффективности параллельной реализации . Это может выполнить набор программ от Intel – Intel Parallel Studio , в котором содержится VTune Amplifier и Intel Parallel Inspector

Intel Cilk Plus – это дополнения языка С/C++, которые используют зарезервированные слова для описания параллелизма и новую систему обозначений для параллельной обработки массивов данных.

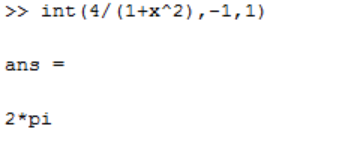
Конструкция cilk\_for предназначена для введения параллелизма в циклы for

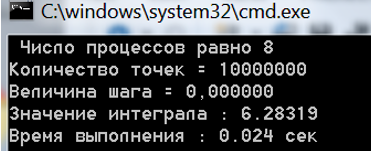
Reducer ы используются для того , чтобы не возникало проблем с общей памятью в параллельных участках кода при выполнении какой – либо операции над общей для всех потоков переменной

Reducer\_opadd используется для суммирования

VTune Amplifier – это профилировщик, необходимый для поиска узких мест в работе программы

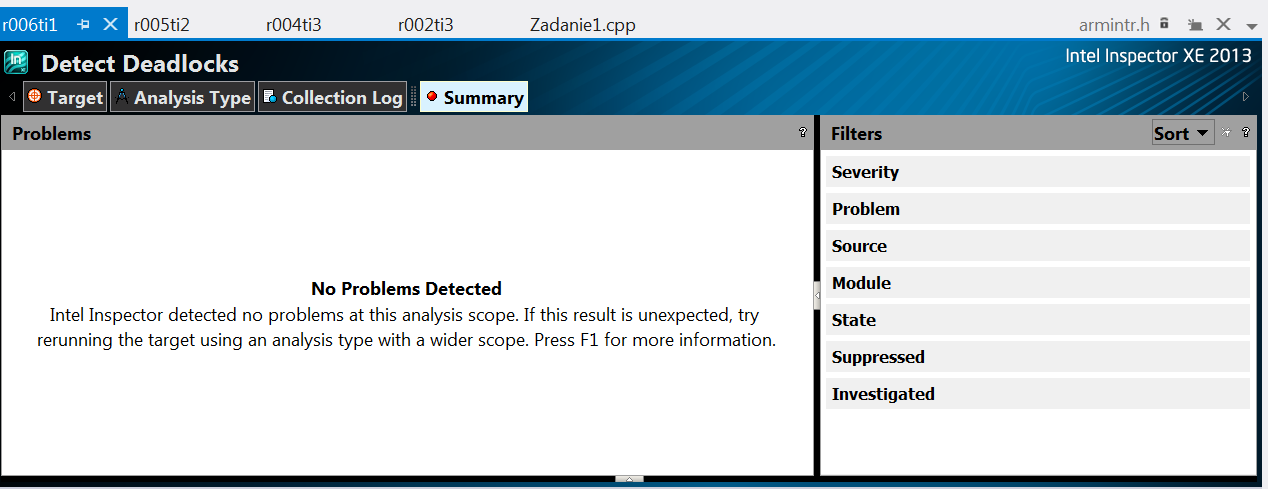
Intel Parallel Inspector – инструмент, предназначенный для тестирования работающей программы с целью выявления основных ошибок, которые возникают при разработке параллельного кода.

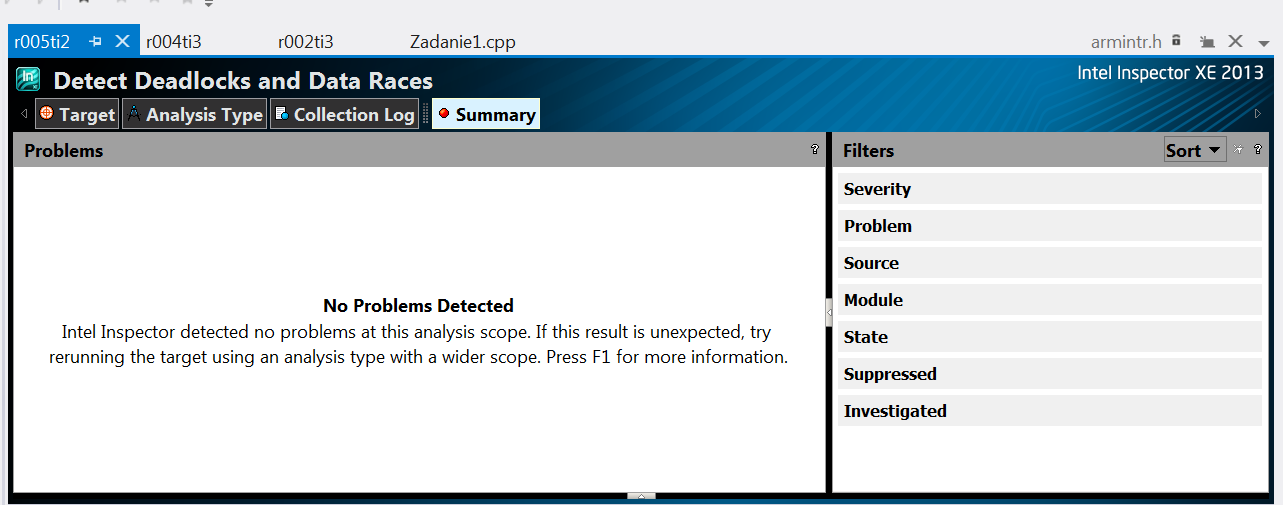


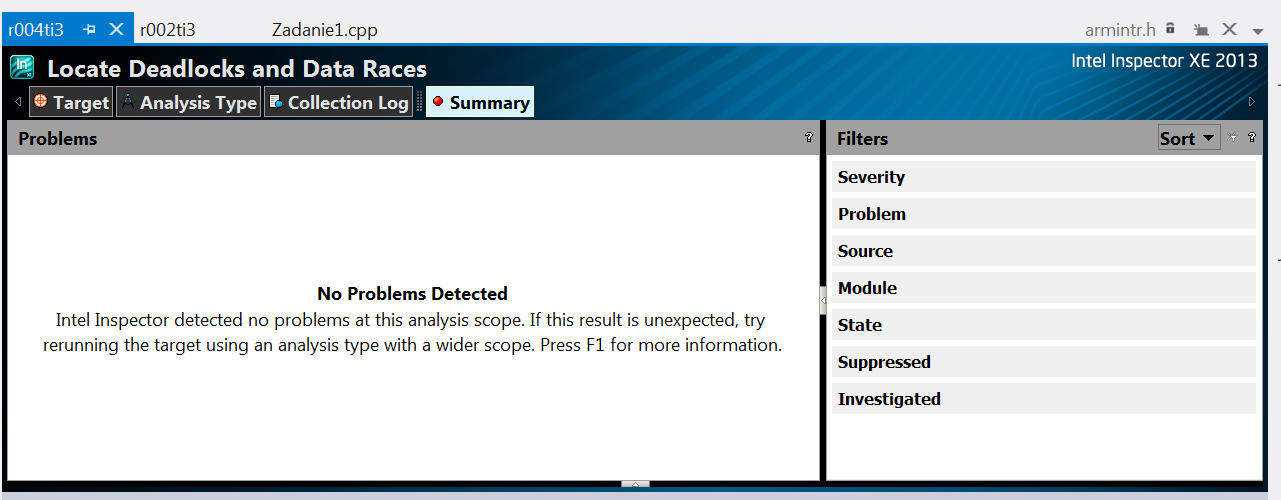


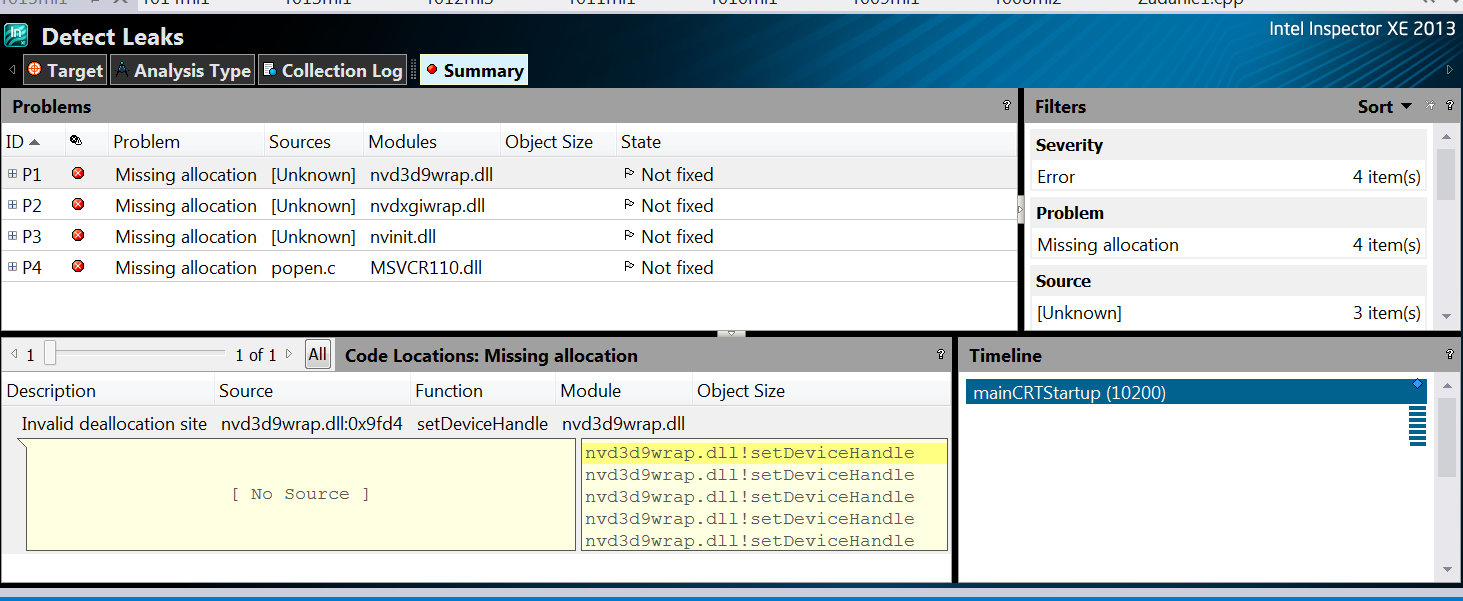
1. Описание и анализ программной реализации

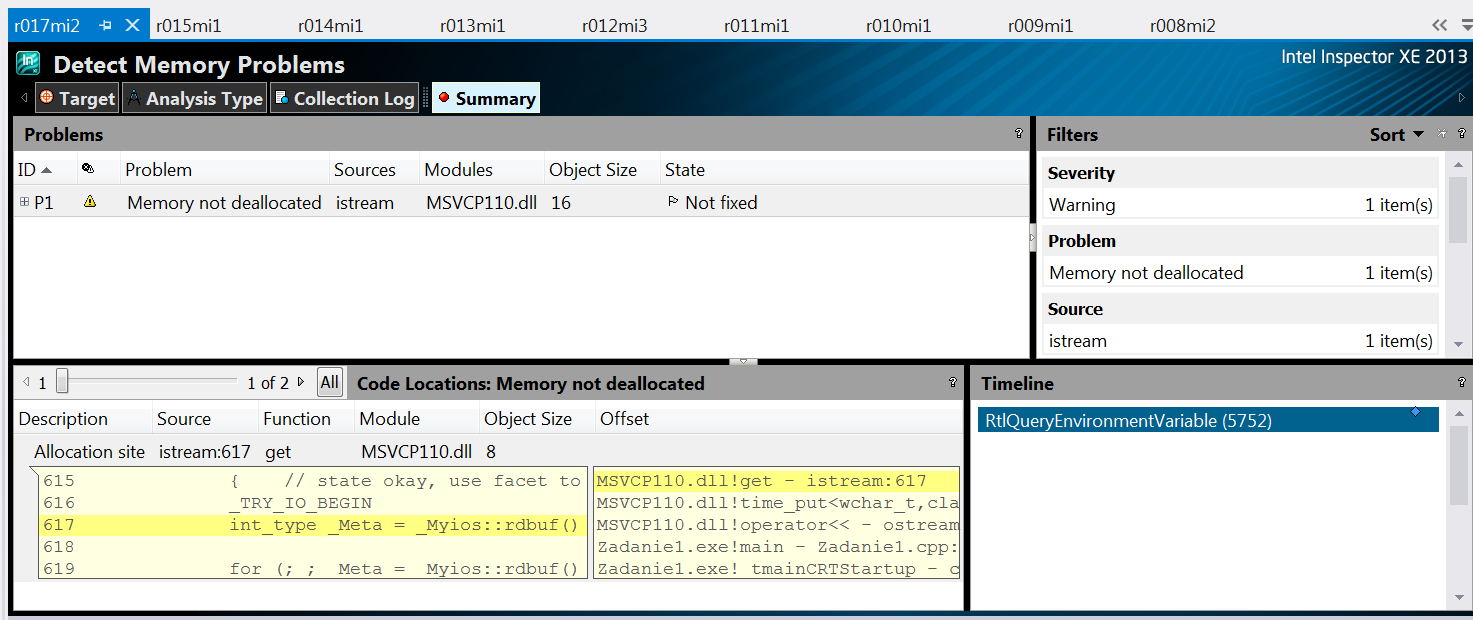
* Анализ работы программы с использованием ***Intel Parallel Inspector XE***;

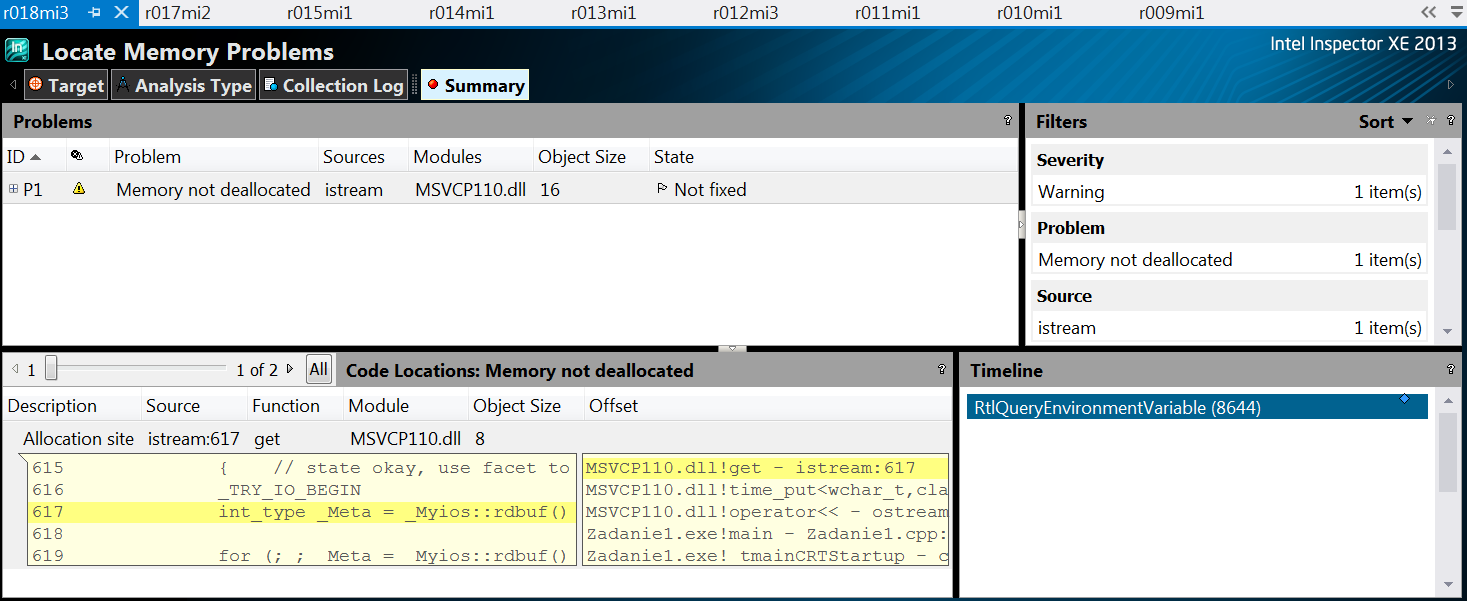




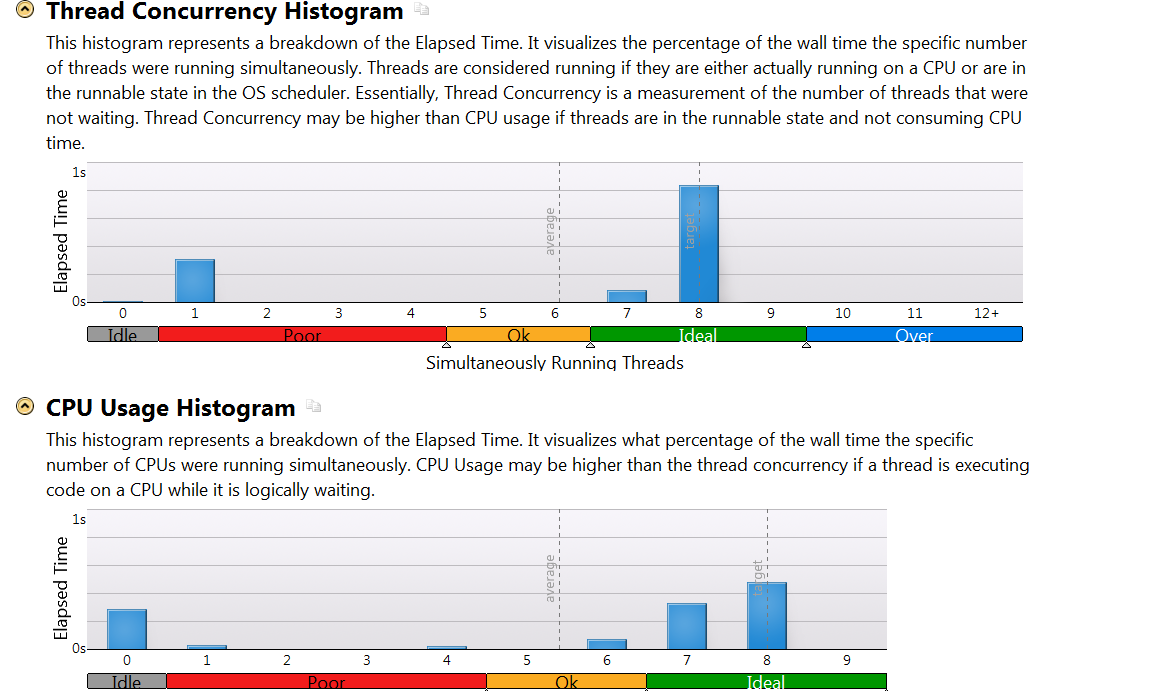


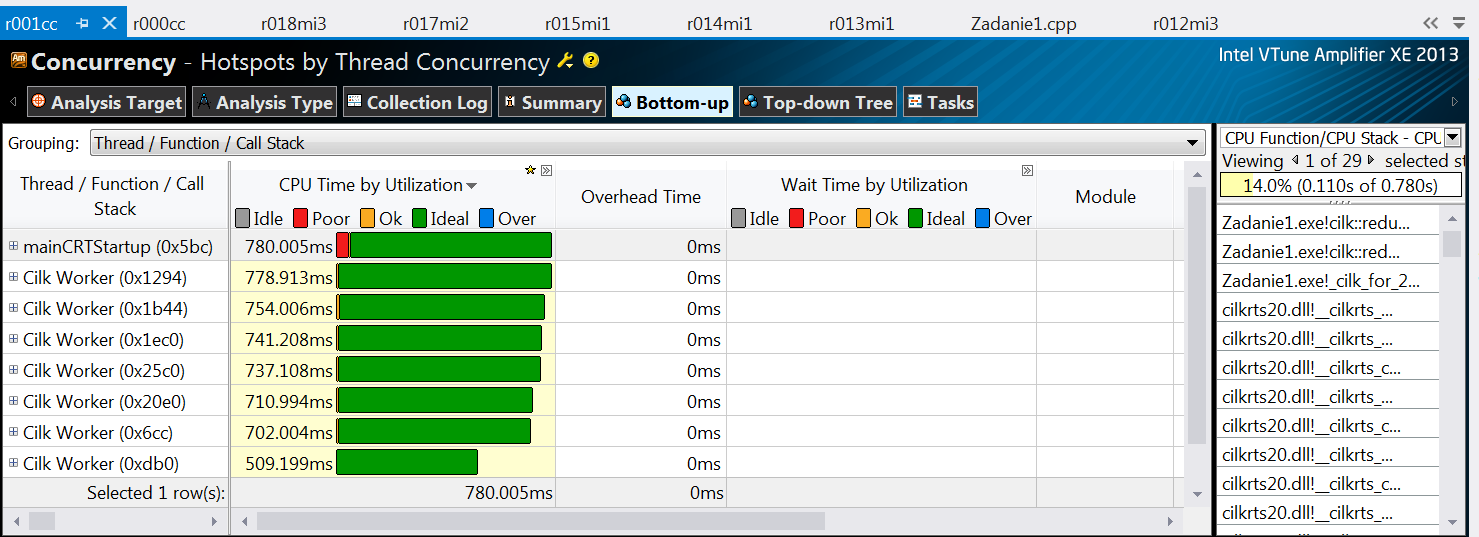


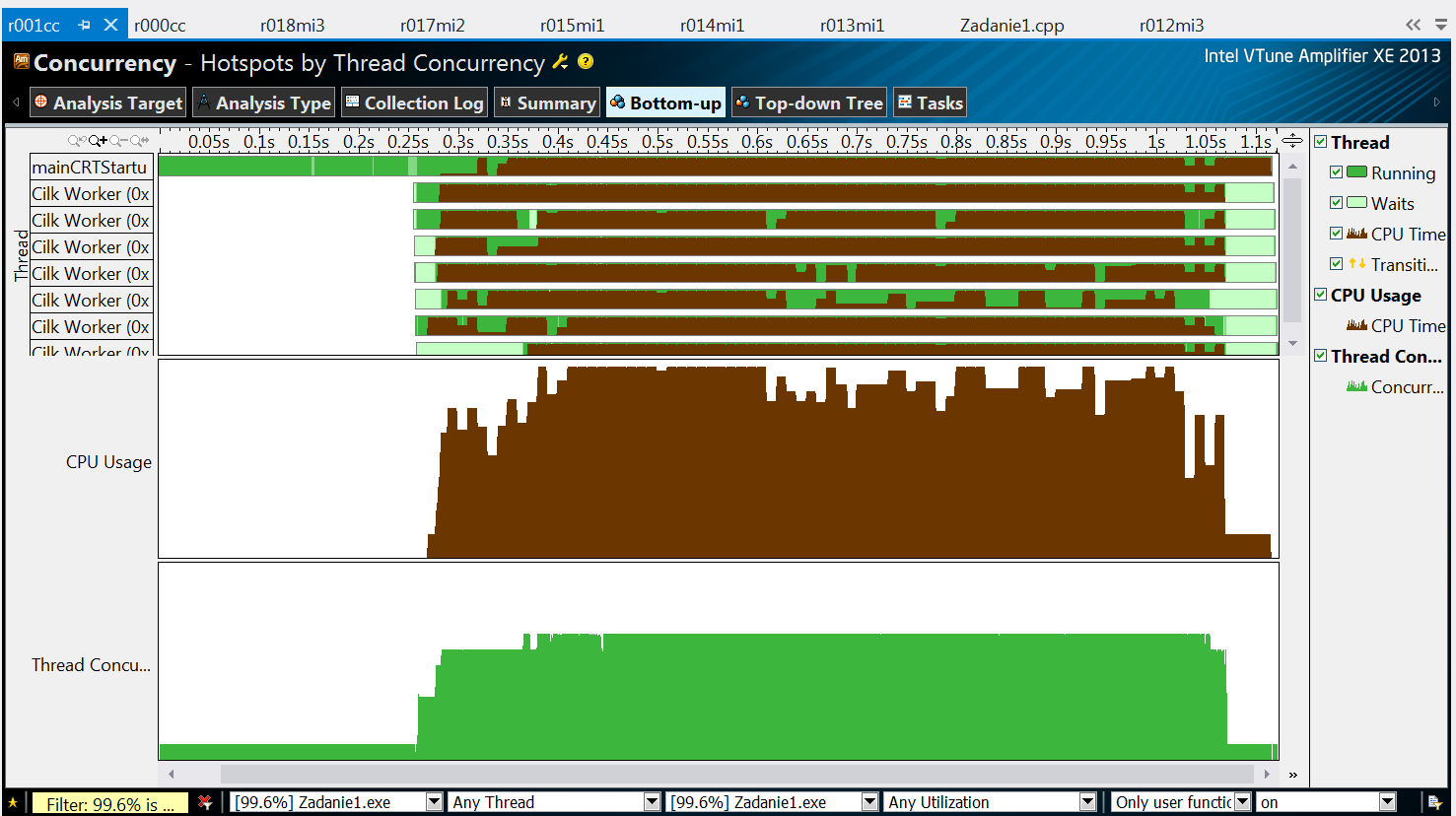


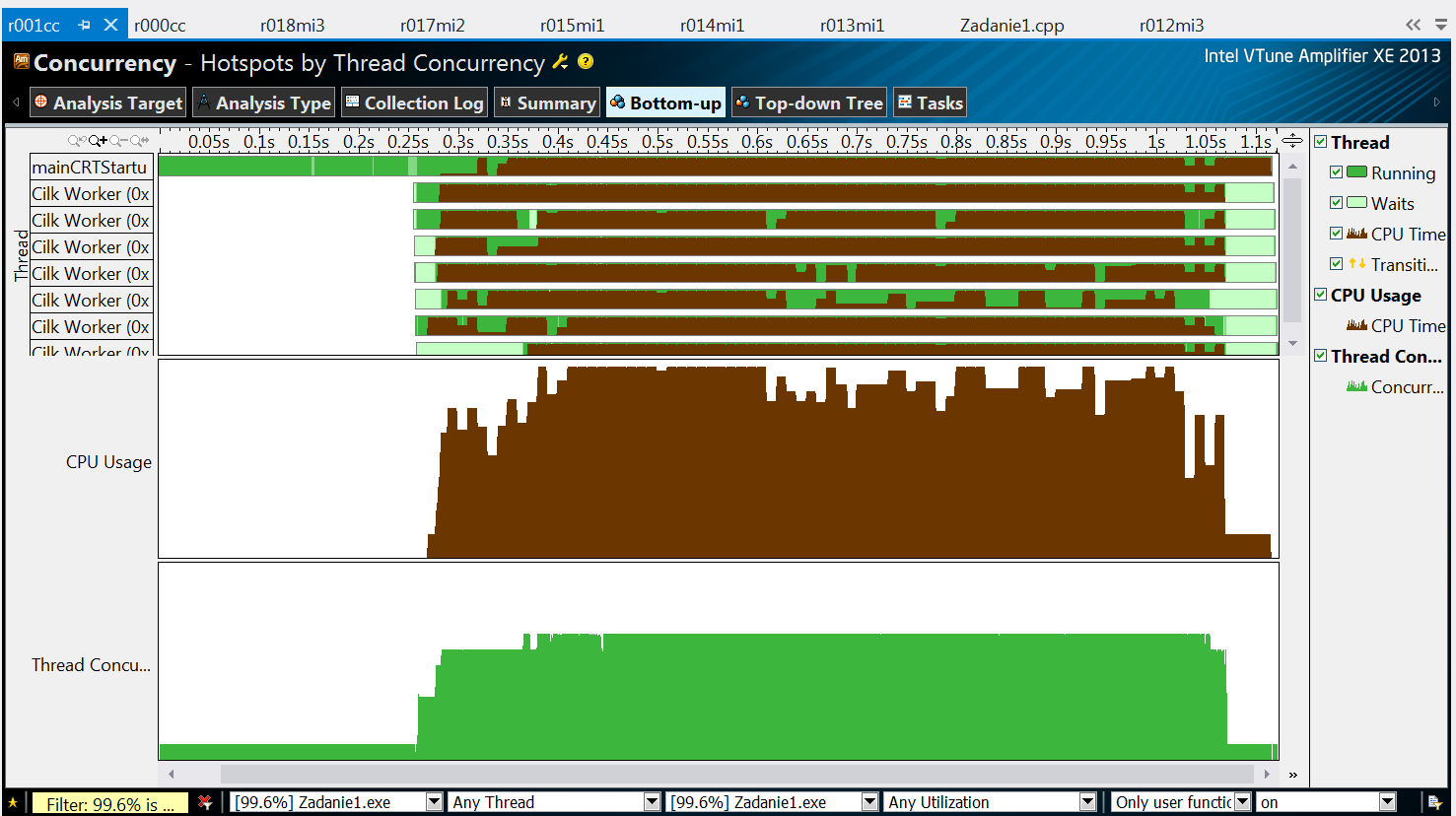


* Оценка эффективности программной реализации;
* Проверка выполнения работы программы с использованием ***Intel VTune Amplifier XE***;

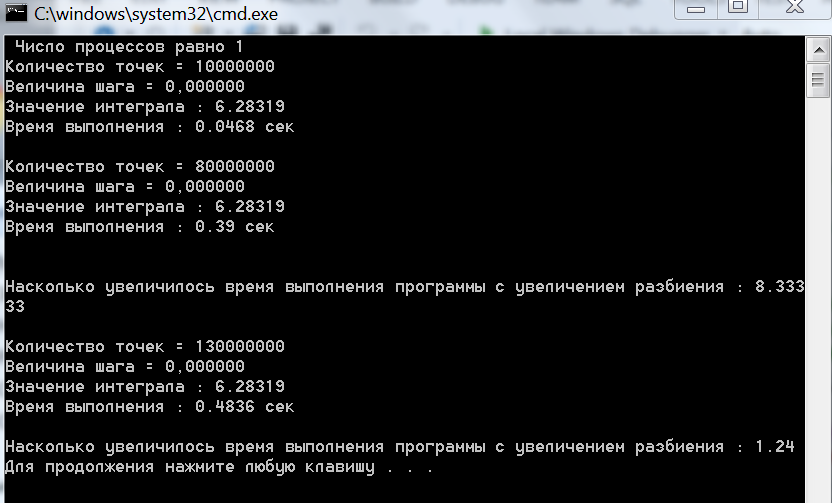


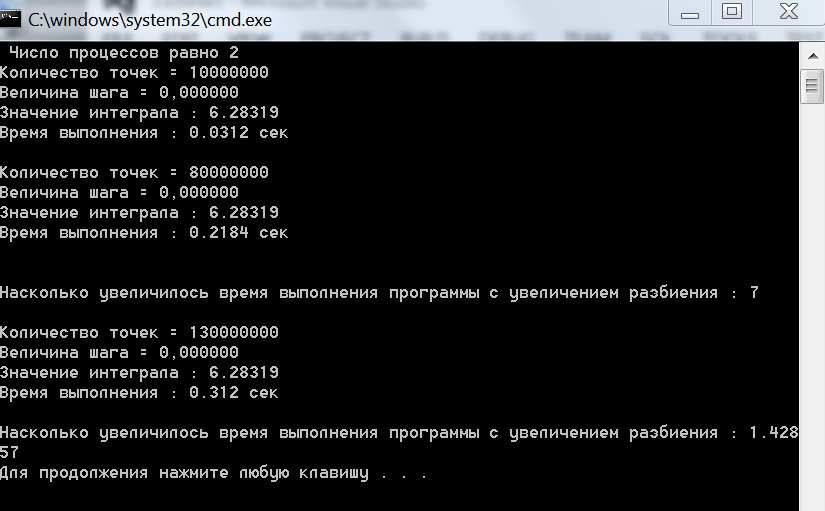


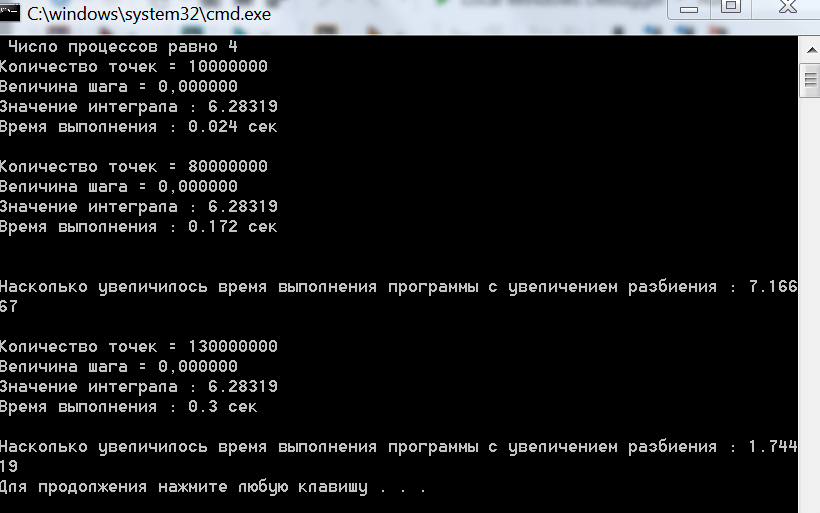


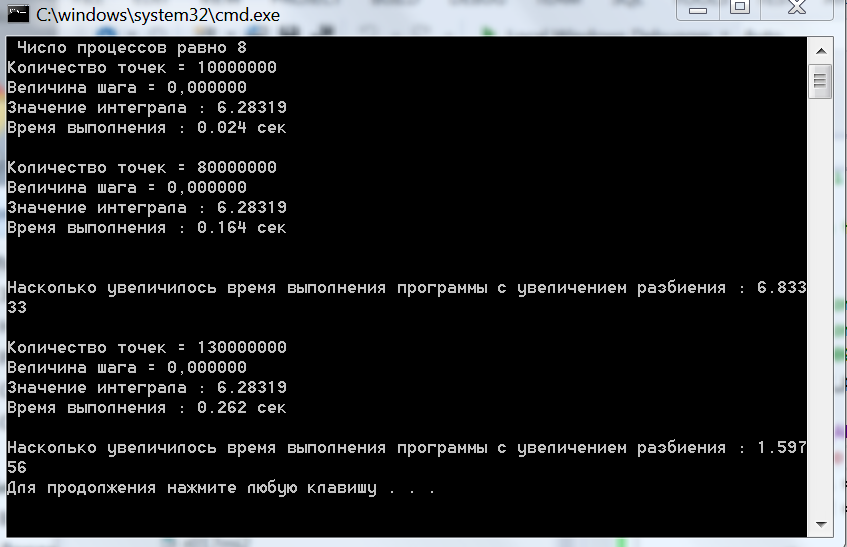


* Сведения о зависимости времени выполнения от заданных параметров алгоритма.









|  |  |  |  |
| --- | --- | --- | --- |
|  | N=10^7 | N=8\*10^7 | N=13\*10^7 |
| P=1 | 0.0468 | 0.39 | 0.4836 |
| P=2 | 0.0312 | 0.2184 | 0.312 |
| P=4 | 0.024 | 0.172 | 0.3 |
| P=8 | 0.024 | 0.164 | 0.262 |

Здесь Р – количество потоков , N – число разбиений

в цикле cilk\_for итерации распределяются между потоками . Сначала каждому потоку “раздают” итерации цикла , затем каждый поток выполняет свои итерации , а потом происходит синхронизация между потоками . Раздача итераций и синхронизация между потоками занимают некоторое небольшое время . Однако при небольшом количестве итераций из-за этого времени последовательная реализация (или реализация с меньшим количесвом потоков ) будет быстрее или примерно таким же (время синхронизации будет велико по сравнению с временем вычислений ) , что мы и видим при 4 и 8 потоках и N=10^7 . Но при больших количествах итераций целесообразнее использовать большее число потоков (время синхронизации будет очень мало по сравнению с временем вычислений ) .