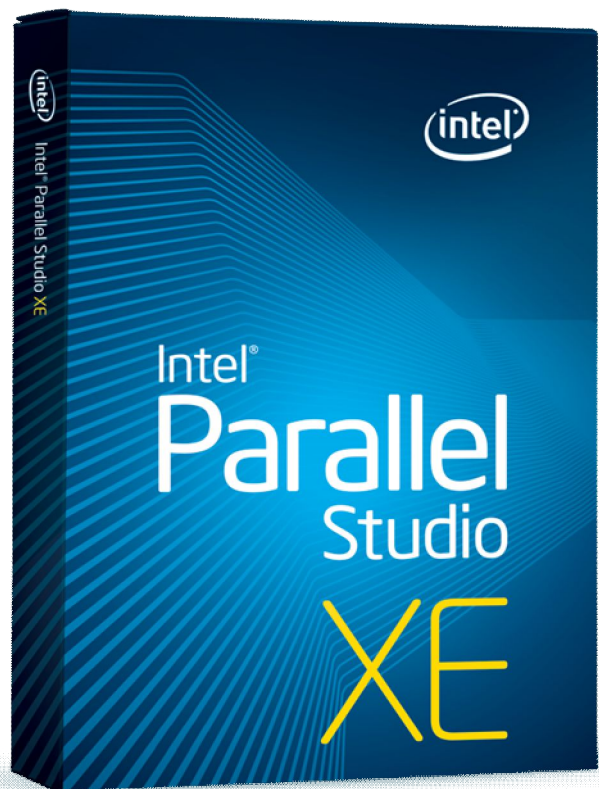


Лекция № 4
Инструменты Intel для оптимизации
отладки приложений на платформе
Windows



Intel Parallel Studio

Parallel Studio – набор инструментов параллельного
программирования разработчикам C/C++ для
Microsoft Visual Studio

Intel Parallel Advisor

поиск в коде участков,
эффективнее
работающих при
использовании
параллельной
обработки

Intel Parallel Composer

компилятор C/C++ и широкий
набор библиотек для
многопоточных алгоритмов

Intel Parallel Inspector

поиск ошибок в коде
приложений для любых
моделей параллельного
программирования

Intel Parallel Amplifier

анализ производительности
приложений для поиска
узких мест в алгоритме

Пример: поиск подстроки

a b a b b a b

Рассмотрим подстроки первого символа:

a|b a b b a b

a|b a b b a b

Увеличиваем подстроку:

a|b a b b a b

a|b a b b a b

Увеличиваем подстроку:

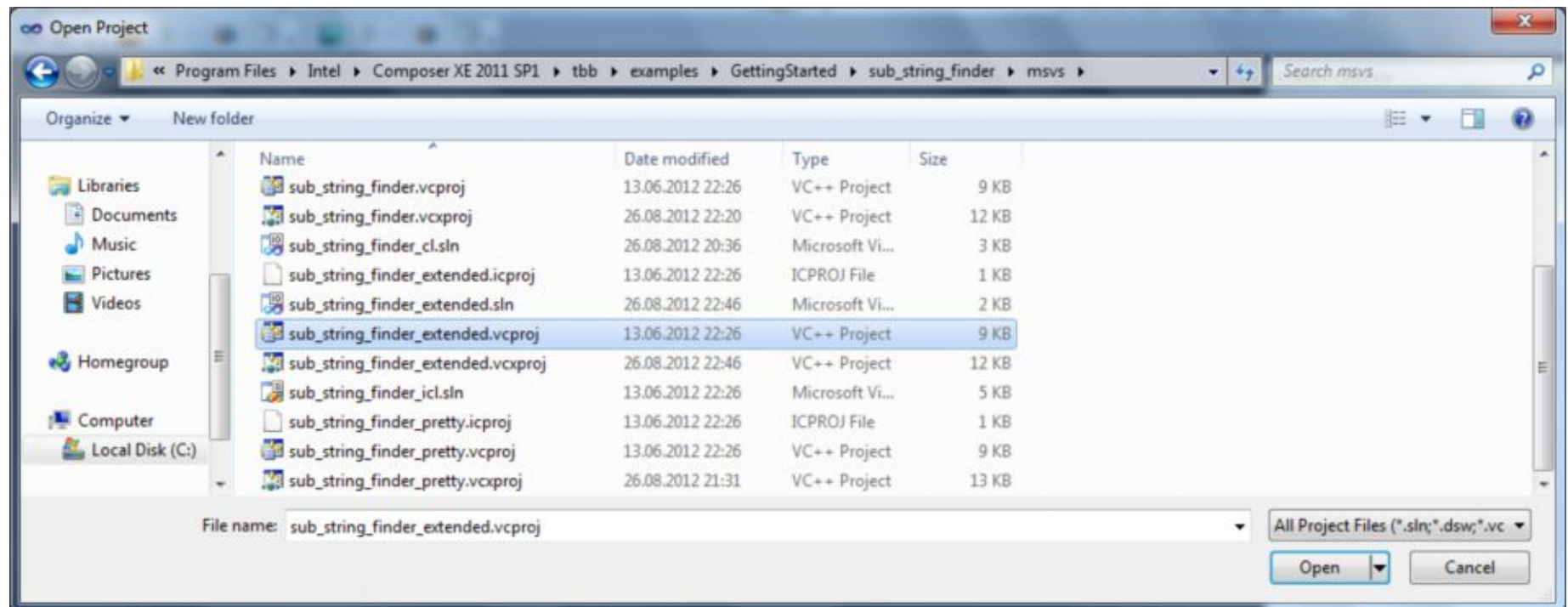
a|b a b b a b

Получили, что максимальная подстрока для первого символа, которая встречается где-либо еще, длины 2

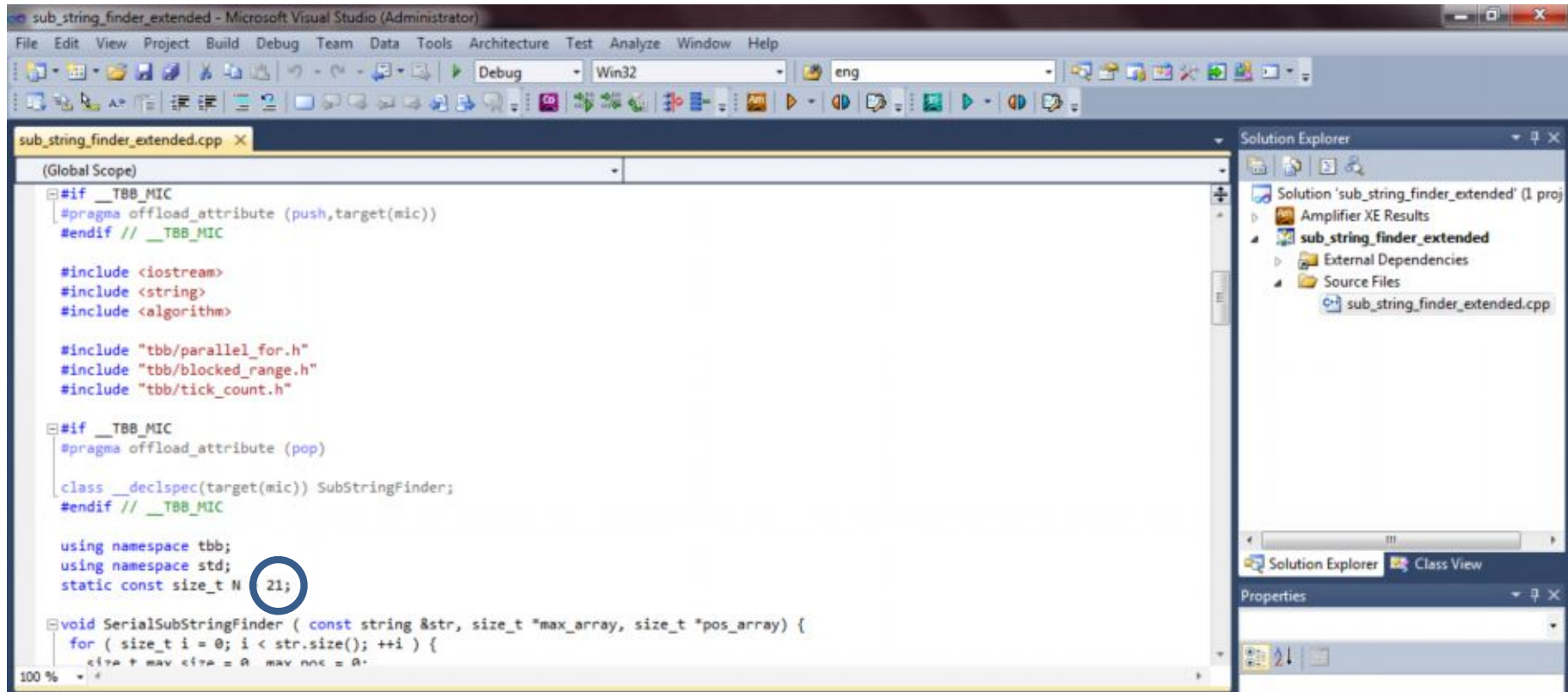
Пример: поиск подстроки

sub_string_finder_extended.vcproj

Program Files\Intel\Composer XE 2011 SP11\
tbb\examples\sub_string_finder\msvs\



Пример: поиск подстроки



```
sub_string_finder_extended.cpp
(Global Scope)
#include <TBB_MIC>
#pragma offload_attribute (push,target(mic))
#endif // __TBB_MIC

#include <iostream>
#include <string>
#include <algorithm>

#include "tbb/parallel_for.h"
#include "tbb/blocked_range.h"
#include "tbb/tick_count.h"

#ifdef __TBB_MIC
#pragma offload_attribute (pop)

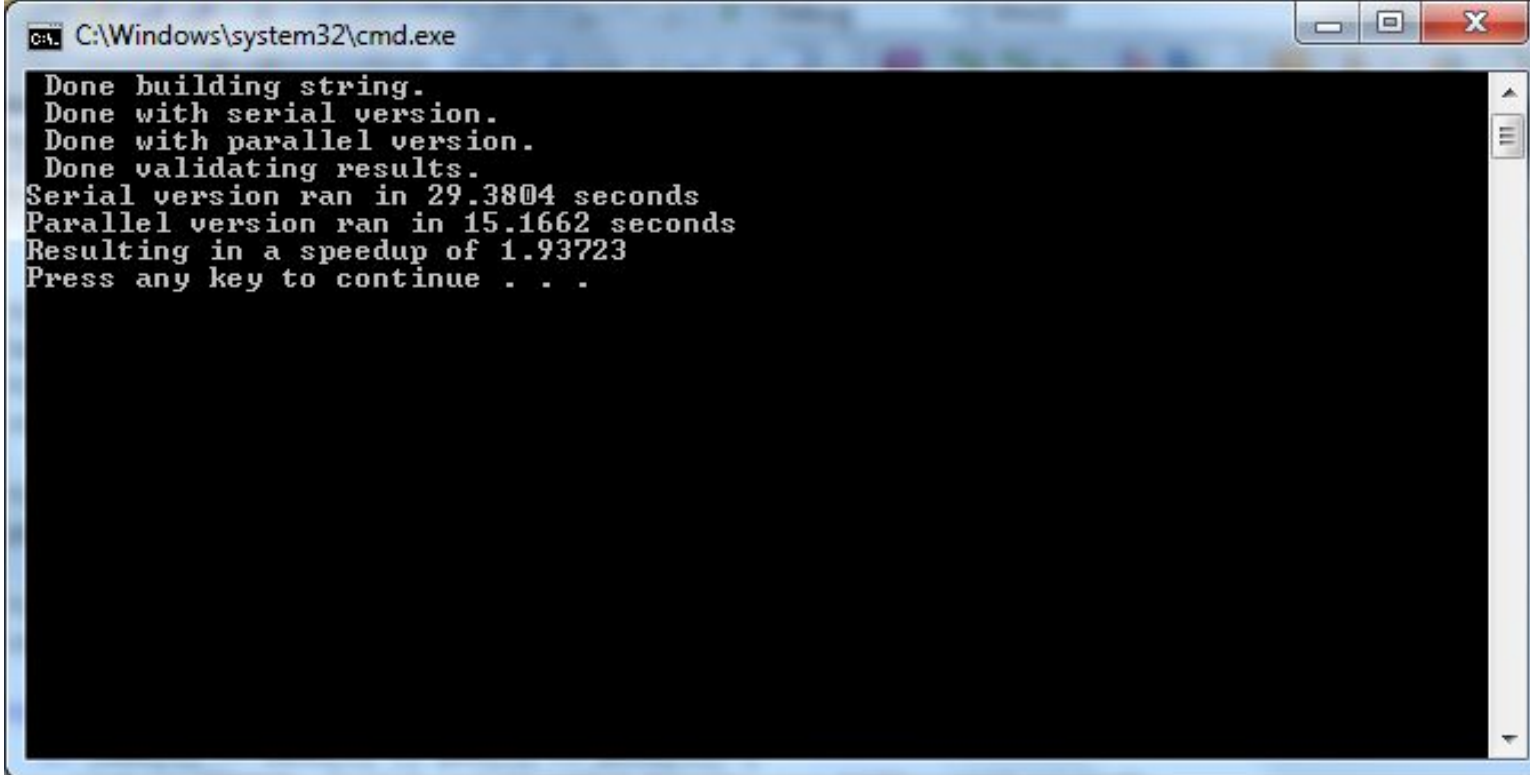
class __declspec(target(mic)) SubStringFinder;
#endif // __TBB_MIC

using namespace tbb;
using namespace std;
static const size_t N = 21;

void SerialSubStringFinder ( const string &str, size_t *max_array, size_t *pos_array ) {
    for ( size_t i = 0; i < str.size(); ++i ) {
        if ( str[i] == 'a' ) *max_array = i;
    }
}
```

Совет: в зависимости от мощности компьютера следует изменить число итераций при генерации строки

Пример: поиск подстроки

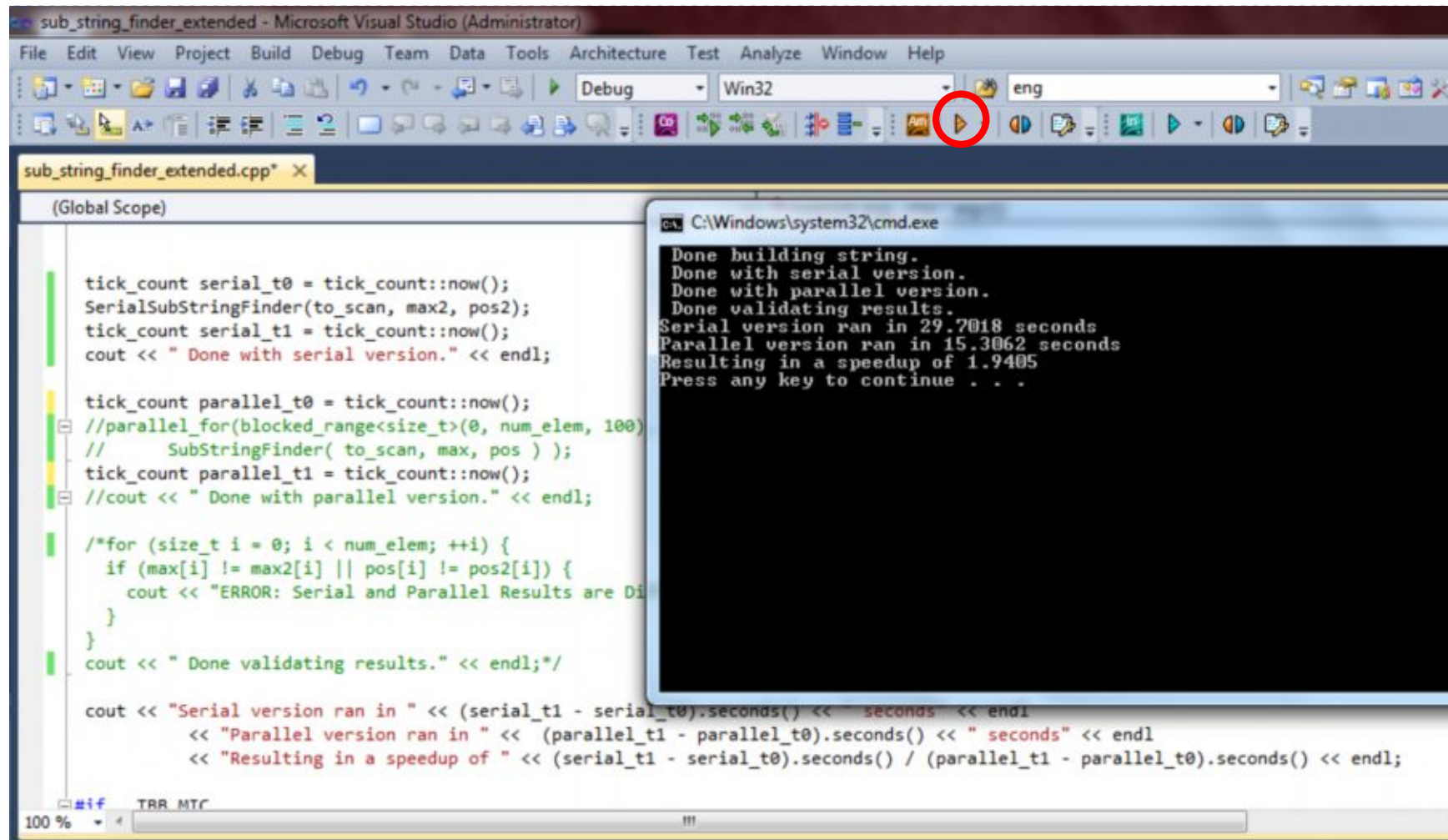


```
C:\Windows\system32\cmd.exe
Done building string.
Done with serial version.
Done with parallel version.
Done validating results.
Serial version ran in 29.3804 seconds
Parallel version ran in 15.1662 seconds
Resulting in a speedup of 1.93723
Press any key to continue . . .
```

Алгоритм реализован двумя способами:

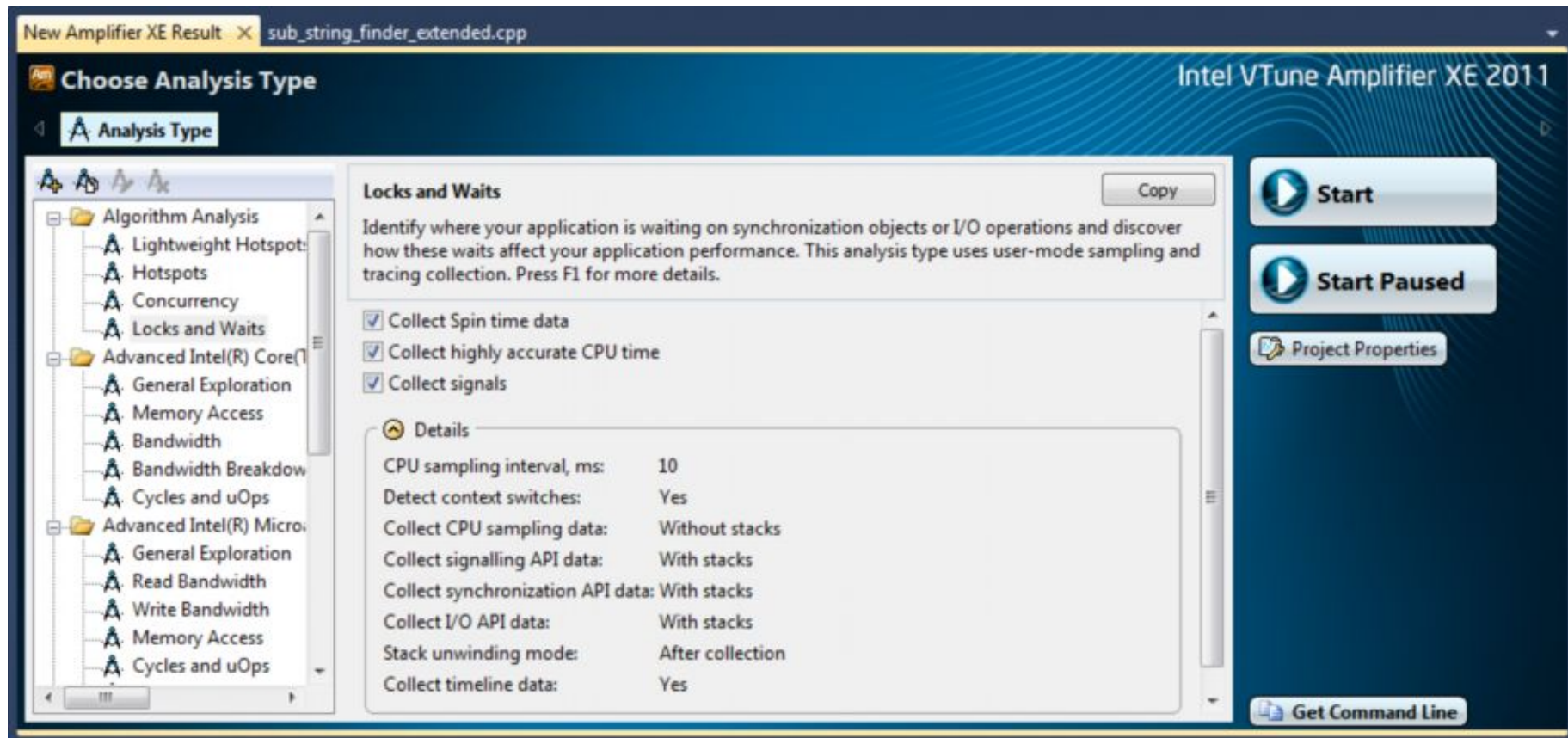
- ▶ Serial Sub String
- ▶ Sub String Finder (с использованием цикла `parallel_for`)

Работа с Amplifier



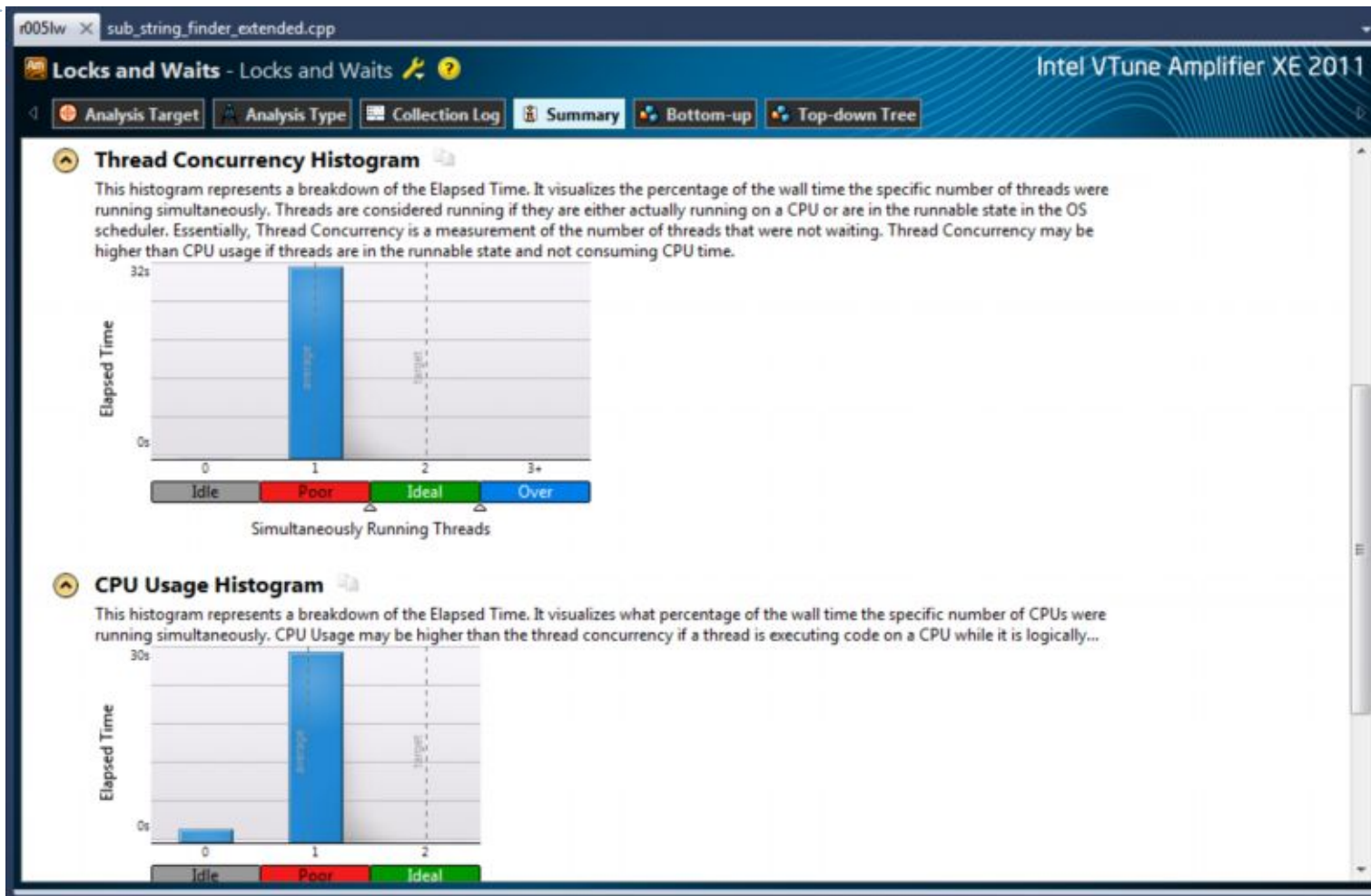
Проводим тестирование для способа Serial Sub String

Работа с Amplifier



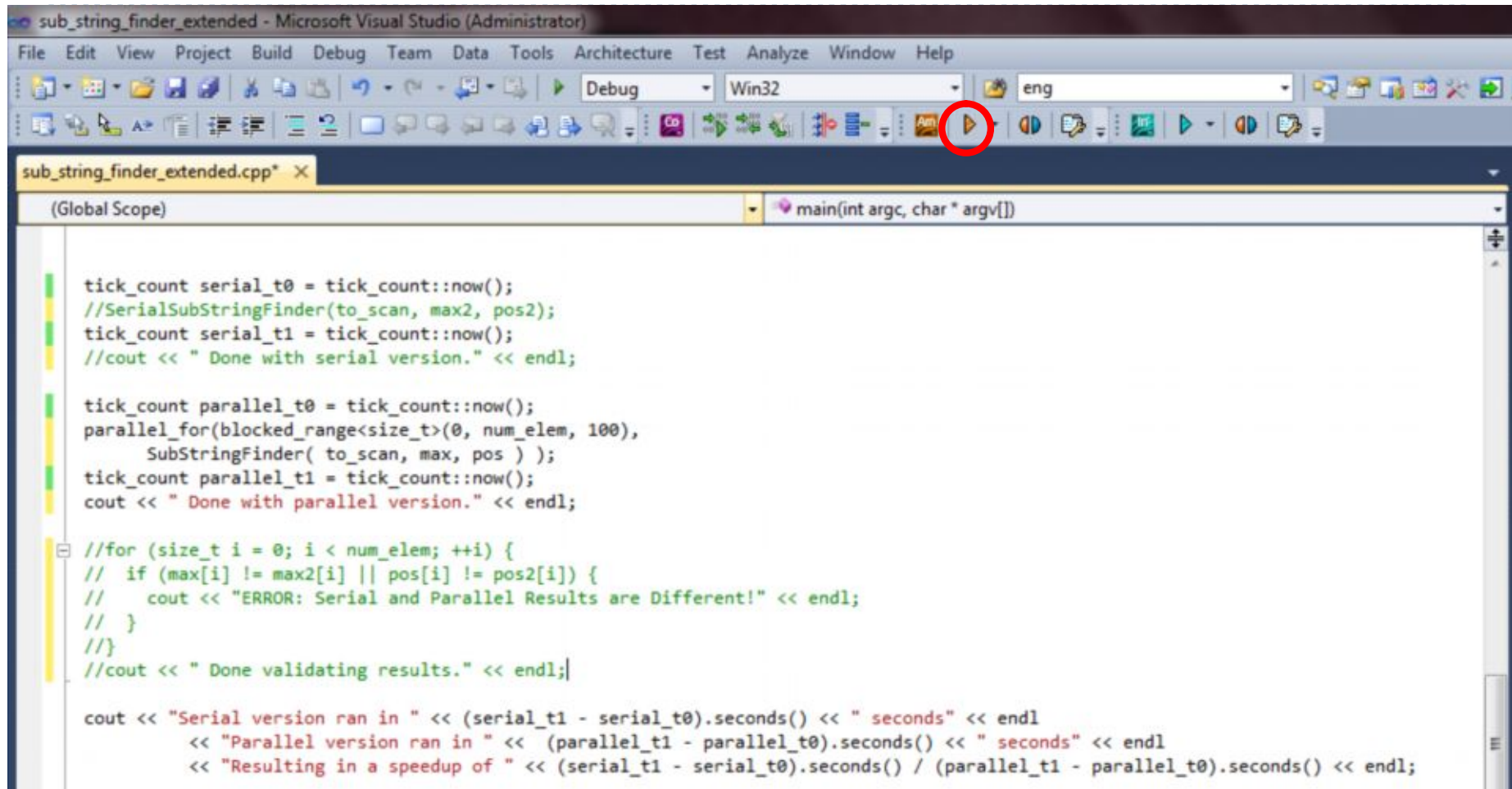
Выбираем тип тестирования
Locks and Waits

Работа с Amplifier



Результаты тестирования для способа Serial Sub String

Работа с Amplifier



The screenshot shows the Microsoft Visual Studio (Administrator) interface. The title bar reads "sub_string_finder_extended - Microsoft Visual Studio (Administrator)". The menu bar includes File, Edit, View, Project, Build, Debug, Team, Data, Tools, Architecture, Test, Analyze, Window, and Help. The toolbar contains various icons, with the Amplifier icon (a blue square with a white 'A') circled in red. The main editor window displays the source code for "sub_string_finder_extended.cpp" in the "main(int argc, char * argv[])" scope. The code includes timing and validation logic for a SubStringFinder function, comparing serial and parallel execution times.

```
tick_count serial_t0 = tick_count::now();
//SerialSubStringFinder(to_scan, max2, pos2);
tick_count serial_t1 = tick_count::now();
//cout << " Done with serial version." << endl;

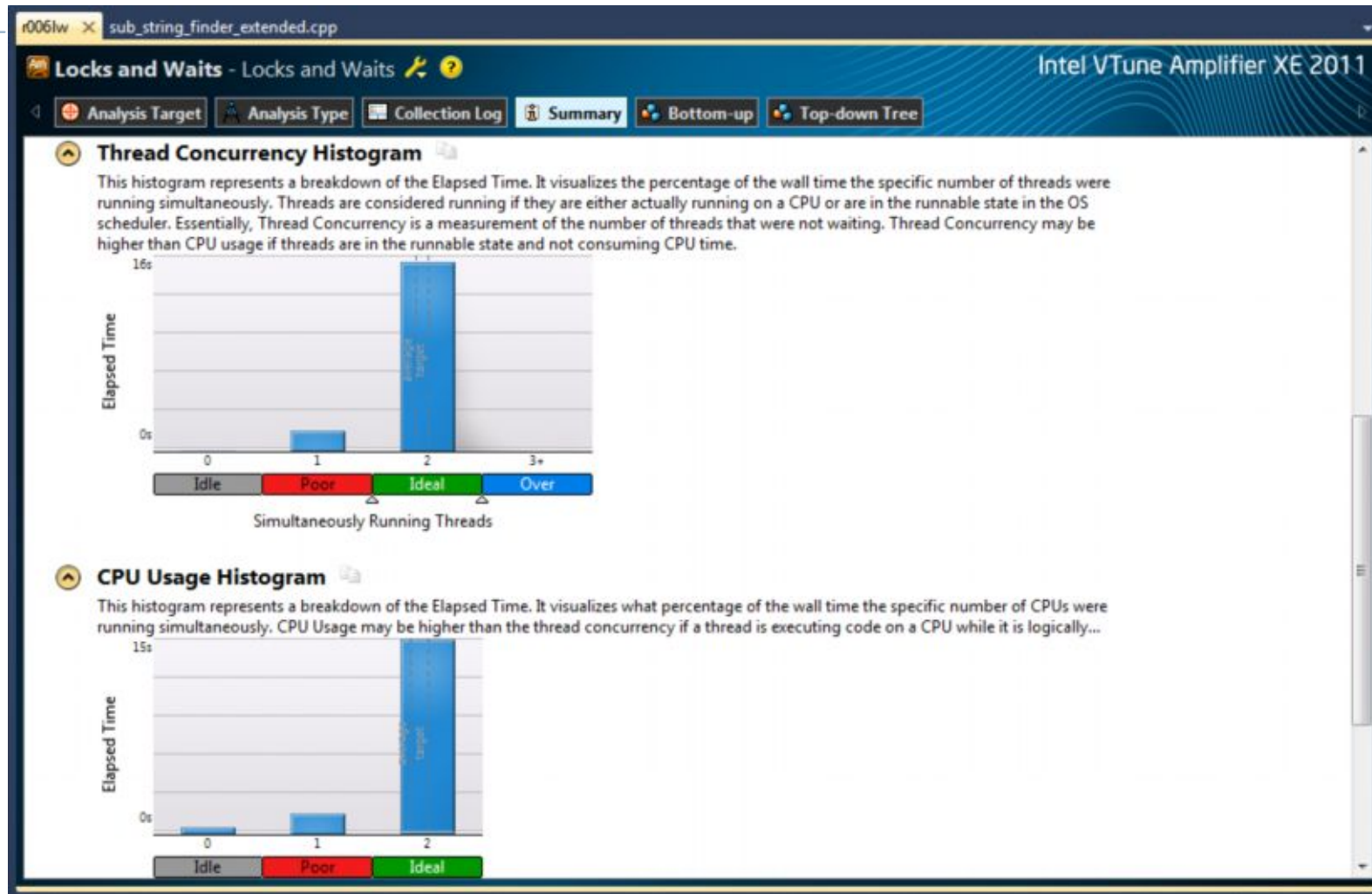
tick_count parallel_t0 = tick_count::now();
parallel_for(blocked_range<size_t>(0, num_elem, 100),
    SubStringFinder( to_scan, max, pos ) );
tick_count parallel_t1 = tick_count::now();
cout << " Done with parallel version." << endl;

//for (size_t i = 0; i < num_elem; ++i) {
// if (max[i] != max2[i] || pos[i] != pos2[i]) {
//     cout << "ERROR: Serial and Parallel Results are Different!" << endl;
// }
//}
//cout << " Done validating results." << endl;

cout << "Serial version ran in " << (serial_t1 - serial_t0).seconds() << " seconds" << endl
    << "Parallel version ran in " << (parallel_t1 - parallel_t0).seconds() << " seconds" << endl
    << "Resulting in a speedup of " << (serial_t1 - serial_t0).seconds() / (parallel_t1 - parallel_t0).seconds() << endl;
```

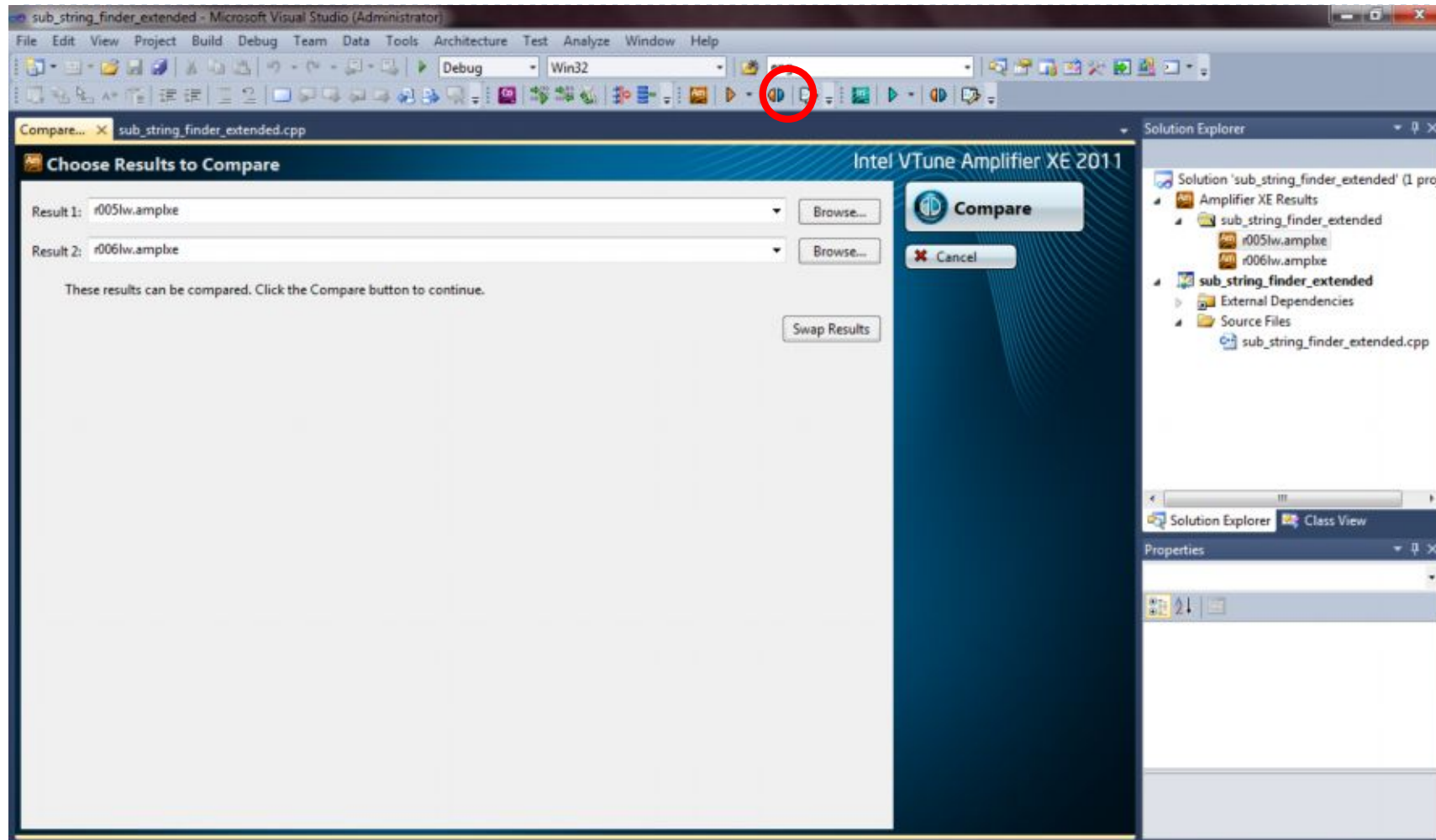
Аналогичным образом тестируем способ Sub String Finder

Работа с Amplifier



Результаты тестирования для способа Sub String Finder

Работа с Amplifier



Сравнение результатов двух способов

Работа с Amplifier



СПбГУ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ



Благодарю за внимание!

Вопросы?