ОПТИМИЗАЦИЯ КОДА

С.А.Немнюгин

Санкт-Петербургский Государственный Университет

s.nemnyugin@spbu.ru

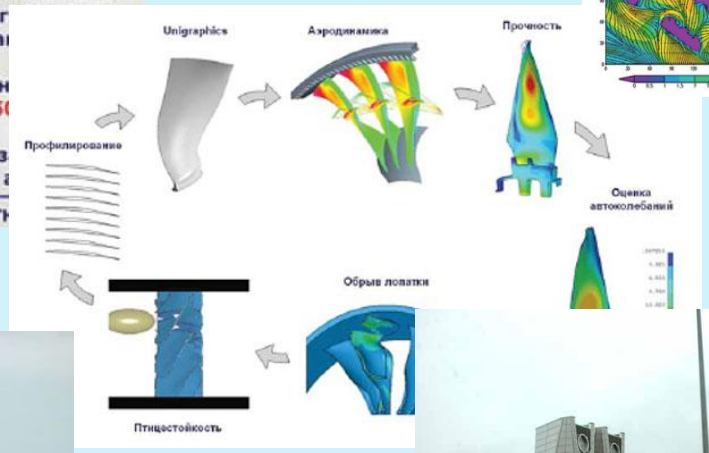
nemnyugin@parserplus.com

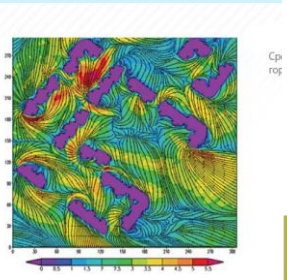
Наша цель?

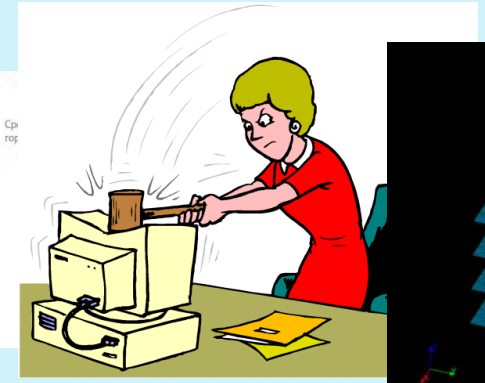
- Решать более сложные задачи, используя более точные, более реалистические, следовательно, более сложные модели. Такие модели обладают более высокими прогностическими возможностями.
- Решать стандартные задачи быстрее.

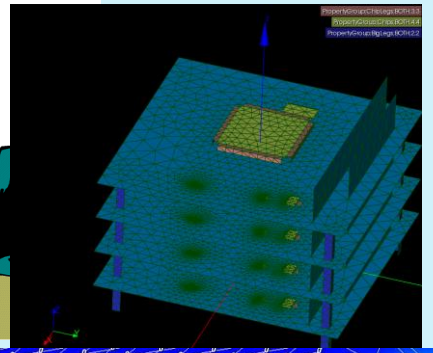
Для чего это нужно?

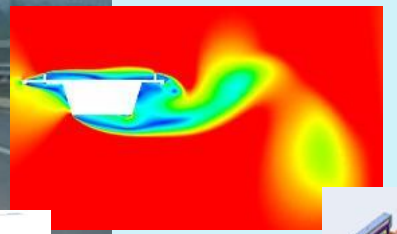

Открыт и запатентован новый класс прямых ингибиторов тромбина – новое лекарство от тромбозов и важный компонент плазмозамещающих растворов
МГУ – дизайн ингибиторов с использованием суперкомпьютеров
ТРОМБИН
На разработку понадобилось около 1,5 лет – обычно: 8 лет
Проводятся испытания на животн

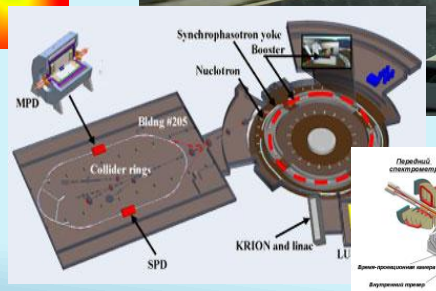

Unigraphics
Аэродинамика
Прочность
Профилирование
Оценка автоколебаний
Обрыв лопатки
Птицестойкость



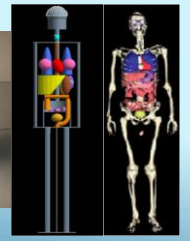


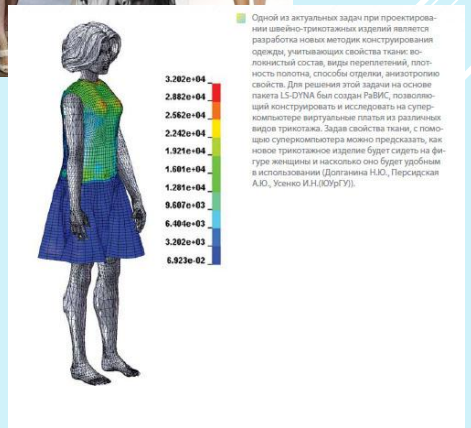





Synchrophasotron yoke
Booster
Nuclotron
MPD
Bldng #205
Collider rings
SPD
KRION and linac


Адронная терапия
в лечении онкологических заболеваний




3.202e+04
2.882e+04
2.562e+04
2.242e+04
1.921e+04
1.601e+04
1.281e+04
9.607e+03
6.404e+03
3.202e+03
6.923e-02


HPC
Платформа
Параллельные технологии
Модель
Оптимизация готовой программы
Алгоритмы
Оптимизация при компиляции
Высокопроизводительные библиотеки
Программирование

1. Разработка приложения построена таким образом, чтобы добиться максимальной производительности.

2. Приложение разработано, необходимо его оптимизировать, увеличив производительность.

Цикл оптимизации

1. Выявление «горячих пятен » (Hotspots).
2. Определение причин низкой эффективности:
   - промахи кэш-памяти;
   - доступ к данным;
   - простои выполнения программы;
   - ошибки предсказания ветвлений;
   - другие.
3. Оптимизация программы, измерение производительности, завершение оптимизации или повторение с п.1.

Три уровня оптимизации:

1. Системный уровень (наибольший потенциал оптимизации).
2. Уровень приложения (умеренный потенциал оптимизации).
3. Микроархитектурный уровень (наименьший потенциал оптимизации).

1 шаг анализа производительности программы - **на уровне системы.**

- Анализ операций обмена с внешними носителями.
- Анализ использования оперативной памяти.
- Анализ взаимодействия с сетью.

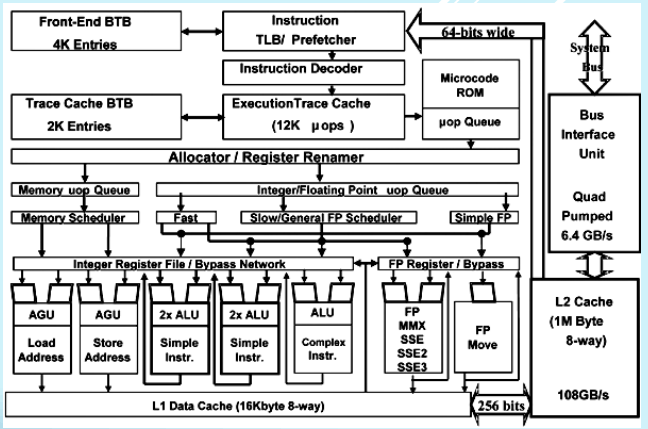
Цель оптимизации на уровне системы – добиться наиболее эффективного взаимодействия программы с системой. Если программа мало загружает процессор из-за проблем на уровне системы, даже серьезное ускорение на уровне архитектуры не даст заметного выигрыша в производительности.

2 шаг - оптимизация **на уровне приложения** (оптимизация алгоритмов).
Выполняется анализ:

- эффективности использования прикладного программного интерфейса;
- эффективности реализации многопоточности;
- наличия блокировок;
- другие.

3 (самый нижний) уровень анализа – уровень архитектуры.
Ключевые факторы:

- использование кэш-памяти;
- выравнивание данных;
- другие вопросы.



Intel® VTune Amplifier XE 2013

Intel® VTune Amplifier XE 2013 – инструмент динамического анализа приложений.

Многоплатформенный: MS Windows (включая Windows Server 2012), Linux.
Архитектуры: Intel® Xeon Phi, Sandy Bridge-EP (Xeon E5), Ivy Bridge (22-нм Core i5 и i7) и другие.
Частичная несовместимость с Hyper V.

Поддержка: C/C++, Fortran, .NET, ассемблер, JAVA, C#.

Интеграция с Microsoft Visual Studio (включая VS 2012).
Возможность работы в режиме командной строки (CLI).
Поддержка автономного интерфейса.
Широкий спектр видов анализа.
Поддержка регрессионного тестирования производительности.
Возможность сравнения результатов тестирования производительности.

Входит в состав следующих пакетов:
- Intel® Parallel Studio XE
- Intel® Cluster Studio XE

Технические детали

Возможны проблемы при одновременном использовании с Hyper V (доступны только базовые виды алгоритмического анализа). Остановить ВМ, выключить Hyper V.

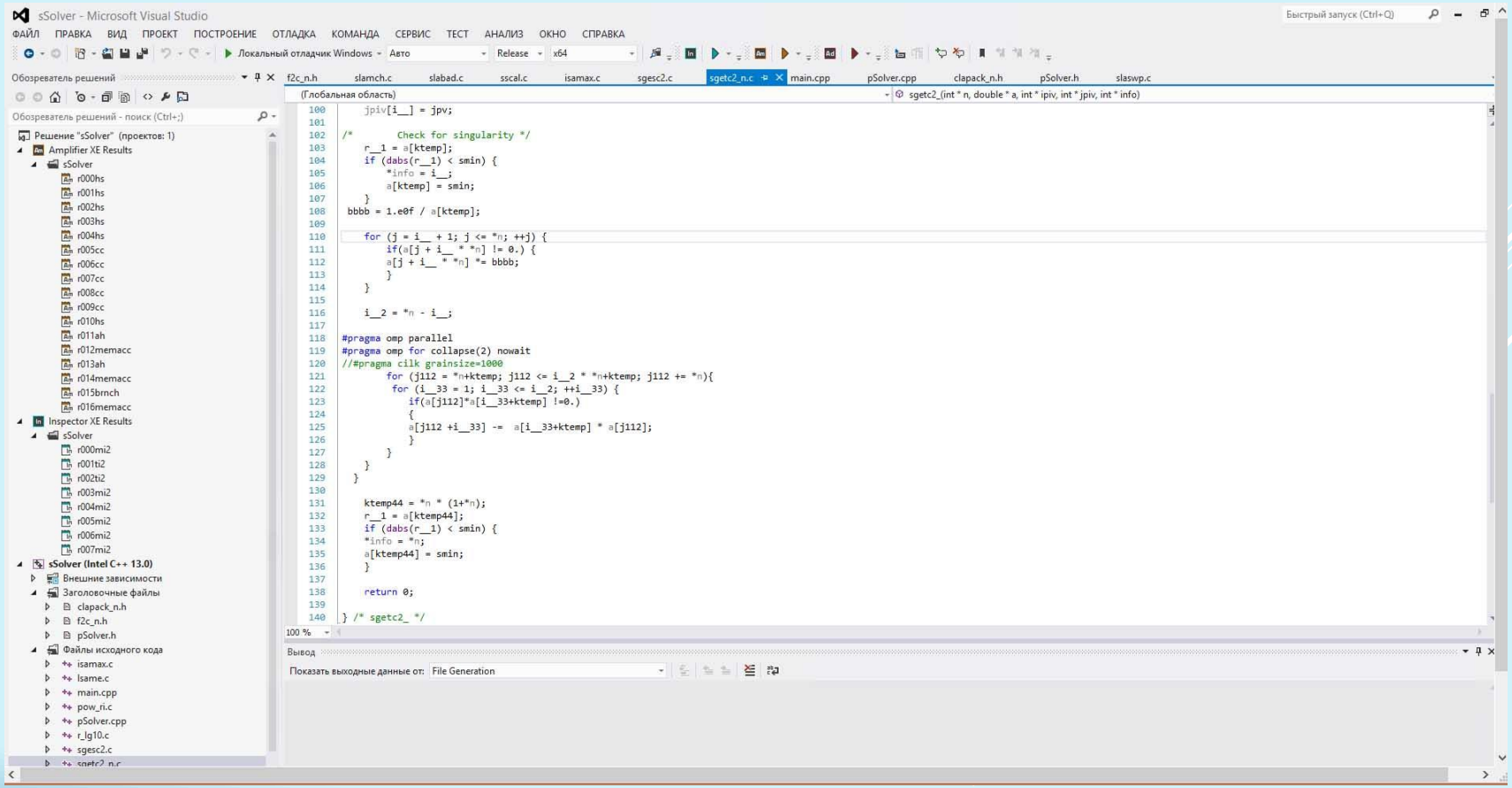
В некоторых ситуациях проблемы могут быть разрешены перезагрузкой драйверов:

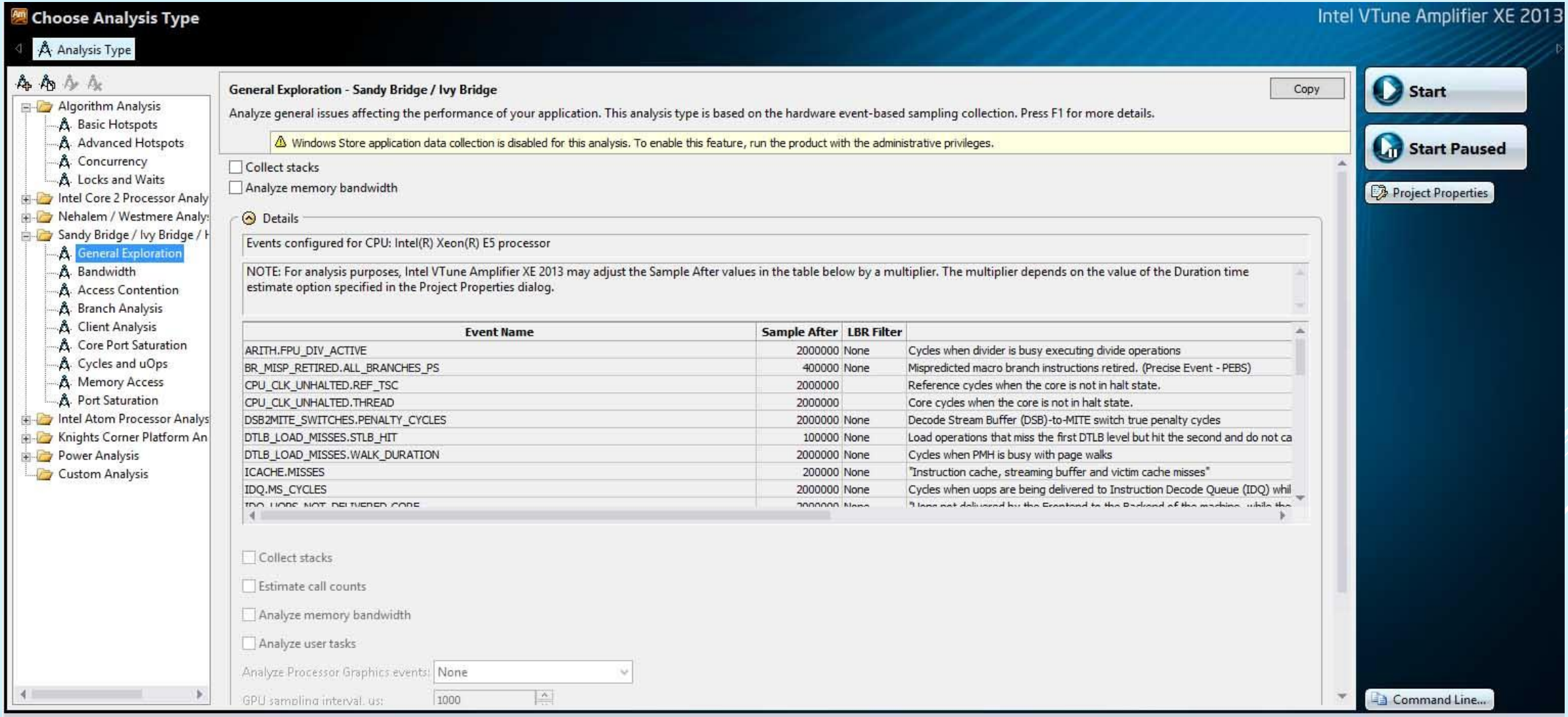
C:\Program Files (x86)\Intel\VTune Amplifier XE 2013\bin32>amplxe-sepreg.exe -u pax
C:\Program Files (x86)\Intel\VTune Amplifier XE 2013\bin32>amplxe-sepreg.exe –i

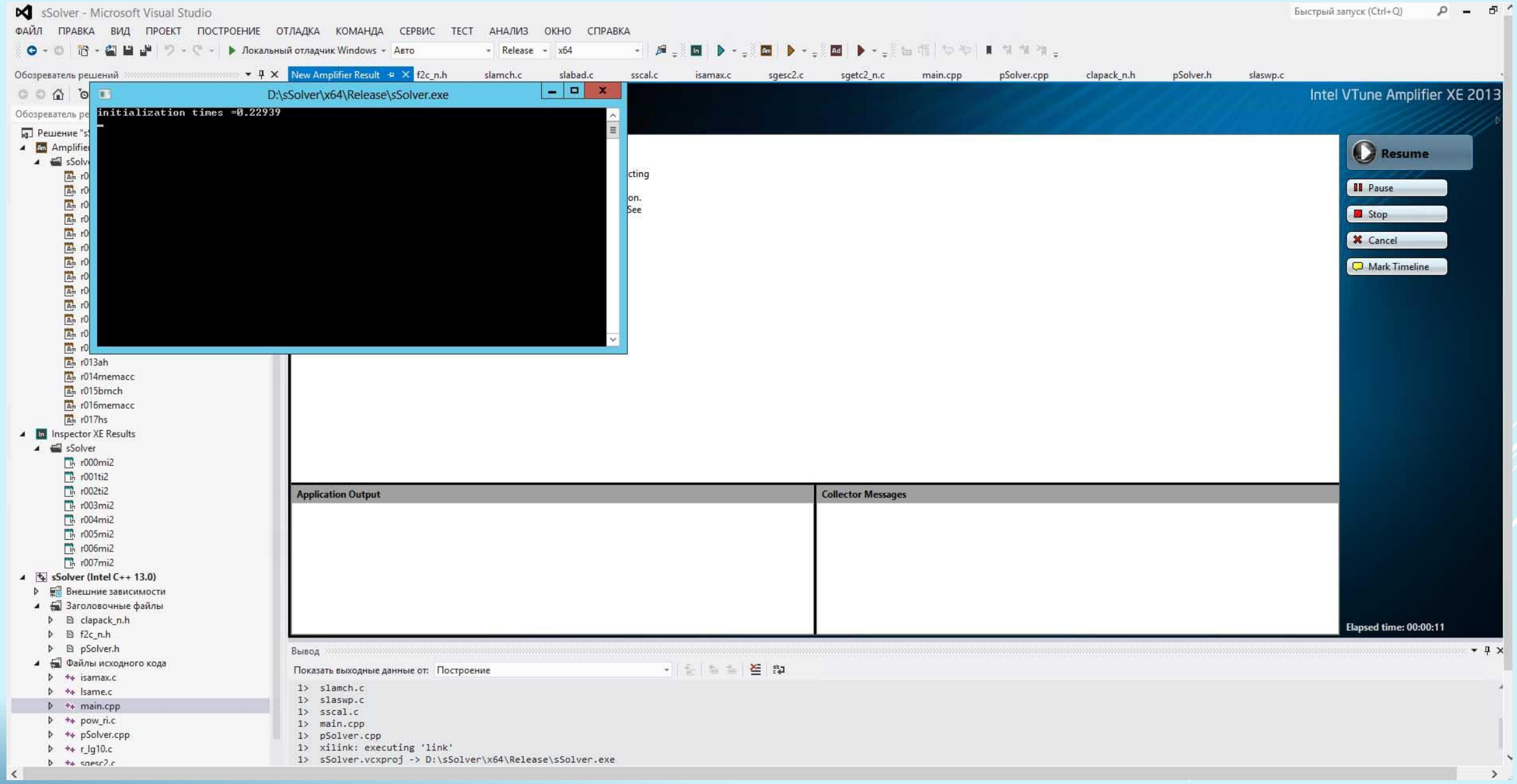
Для выполнения анализа приложения требуется:

- исполняемый (бинарный) файл;
- исходный файл (не обязательно, но желательно, поскольку в этом случае результаты сбора статистики можно «привязать» к исходному коду программы).

Необходим *правильно подготовленный тестовый набор данных*, соответствующий «стандартной» для данного приложения ситуации. Запуск приложения с этим набором данных даёт опорную точку, по которой будет определяться эффективность различных методов оптимизации производительности.




Choose Analysis Type
Intel VTune Amplifier XE 2013
Analysis Type
Algorithm Analysis
Basic Hotspots
Advanced Hotspots
Concurrency
Locks and Waits
Intel Core 2 Processor Analy
Nehalem / Westmere Analy
Sandy Bridge / Ivy Bridge / I
General Exploration
Bandwidth
Access Contention
Branch Analysis
Client Analysis
Core Port Saturation
Cycles and uOps
Memory Access
Port Saturation
Intel Atom Processor Analys
Knights Corner Platform An
Power Analysis
Custom Analysis
General Exploration - Sandy Bridge / Ivy Bridge
Copy
Analyze general issues affecting the performance of your application. This analysis type is based on the hardware event-based sampling collection. Press F1 for more details.
Windows Store application data collection is disabled for this analysis. To enable this feature, run the product with the administrative privileges.
Collect stacks
Analyze memory bandwidth
Details
Events configured for CPU: Intel(R) Xeon(R) E5 processor
NOTE: For analysis purposes, Intel VTune Amplifier XE 2013 may adjust the Sample After values in the table below by a multiplier. The multiplier depends on the value of the Duration time estimate option specified in the Project Properties dialog.
Event Name | Sample After | LBR Filter
ARITH.FPU_DIV_ACTIVE | 2000000 | None | Cycles when divider is busy executing divide operations
BR_MISP_RETIRED.ALL_BRANCHES_PS | 400000 | None | Mispredicted macro branch instructions retired. (Precise Event - PEBS)
CPU_CLK_UNHALTED.REF_TSC | 2000000 | | Reference cycles when the core is not in halt state.
CPU_CLK_UNHALTED.THREAD | 2000000 | | Core cycles when the core is not in halt state.
DSB2MITE_SWITCHES.PENALTY_CYCLES | 2000000 | None | Decode Stream Buffer (DSB)-to-MITE switch true penalty cycles
DTLB_LOAD_MISSES.STLB_HIT | 100000 | None | Load operations that miss the first DTLB level but hit the second and do not ca
DTLB_LOAD_MISSES.WALK_DURATION | 2000000 | None | Cycles when PMH is busy with page walks
ICACHE.MISSES | 200000 | None | "Instruction cache, streaming buffer and victim cache misses"
IDQ.MS_CYCLES | 2000000 | None | Cycles when uops are being delivered to Instruction Decode Queue (IDQ) whil
Collect stacks
Estimate call counts
Analyze memory bandwidth
Analyze user tasks
Analyze Processor Graphics events:
None
GPU sampling interval, us:
1000
Start
Start Paused
Project Properties
Command Line...


sSolver - Microsoft Visual Studio
ФАЙЛ ПРАВКА ВИД ПРОЕКТ ПОСТРОЕНИЕ ОТЛАДКА КОМАНДА СЕРВИС ТЕСТ АНАЛИЗ ОКНО СПРАВКА
Локальный отладчик Windows
Release
x64
D:\sSolver\x64\Release\sSolver.exe
initialization times =0.22939
Intel VTune Amplifier XE 2013
Resume
Pause
Stop
Cancel
Mark Timeline
Application Output
Collector Messages
Elapsed time: 00:00:11
Вывод
Показать выходные данные от: Построение
1> slamch.c
1> slaswp.c
1> sscal.c
1> main.cpp
1> pSolver.cpp
1> xilink: executing 'link'
1> sSolver.vcxproj -> D:\sSolver\x64\Release\sSolver.exe

r018hs | f2c_n.h | slamch.c | slabad.c | sscal.c | isamax.c | sgesc2.c | sgetc2_n.c | main.cpp | pSolver.cpp | clapack_n.h | pSolver.h | slaswp.c

Basic Hotspots Hotspots by CPU Usage viewpoint (change) — Intel VTune Amplifier XE 2013

Analysis Target | Analysis Type | Collection Log | Summary | Bottom-up | Caller/Callee | Top-down Tree | Tasks and Frames

Elapsed Time: 143.905s

Total Thread Count:	16
Overhead Time:	3.893s
Spin Time:	114.607s

A significant portion of CPU time is spent waiting. Use this metric to discover which synchronizations are spinning. Consider adjusting spin wait parameters, changing the lock implementation (for example, by backing off then descheduling), or adjusting the synchronization granularity.

CPU Time:	893.911s
Paused Time:	0s

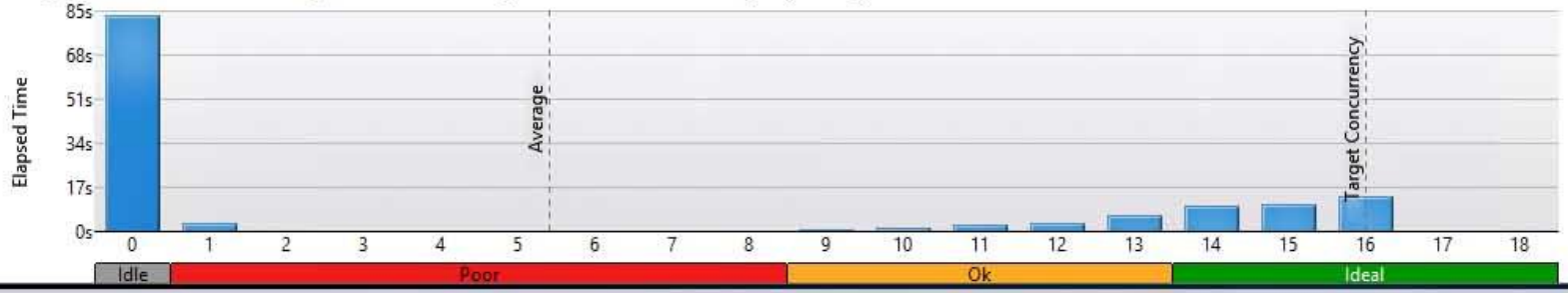
Top Hotspots

This section lists the most active functions in your application. Optimizing these hotspot functions typically results in improving overall application performance.

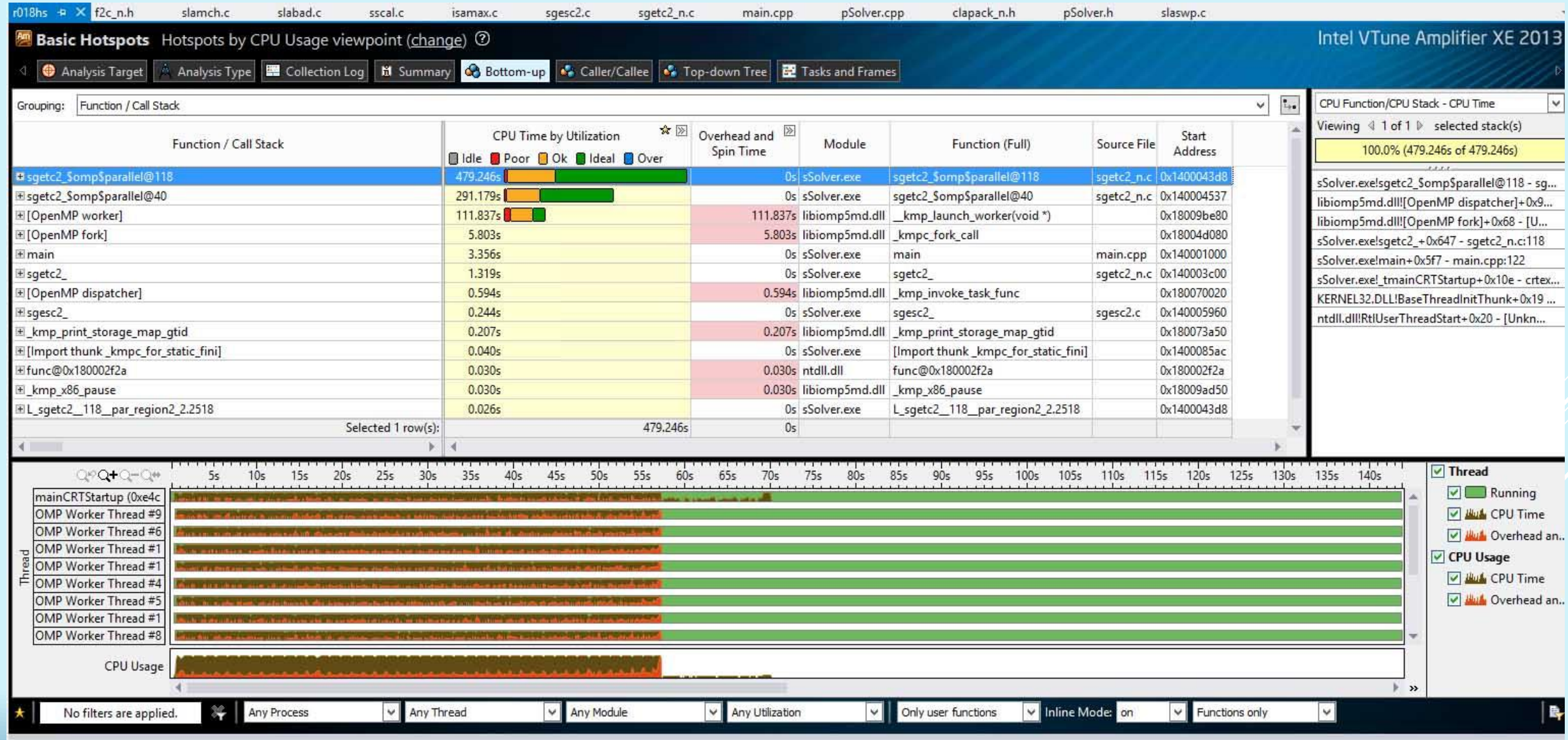
Function	CPU Time
sgetc2_ompparallel@118	479.246s
sgetc2_ompparallel@40	291.179s
[OpenMP worker]	111.837s
[OpenMP fork]	5.803s
main	3.356s
[Others]	2.489s

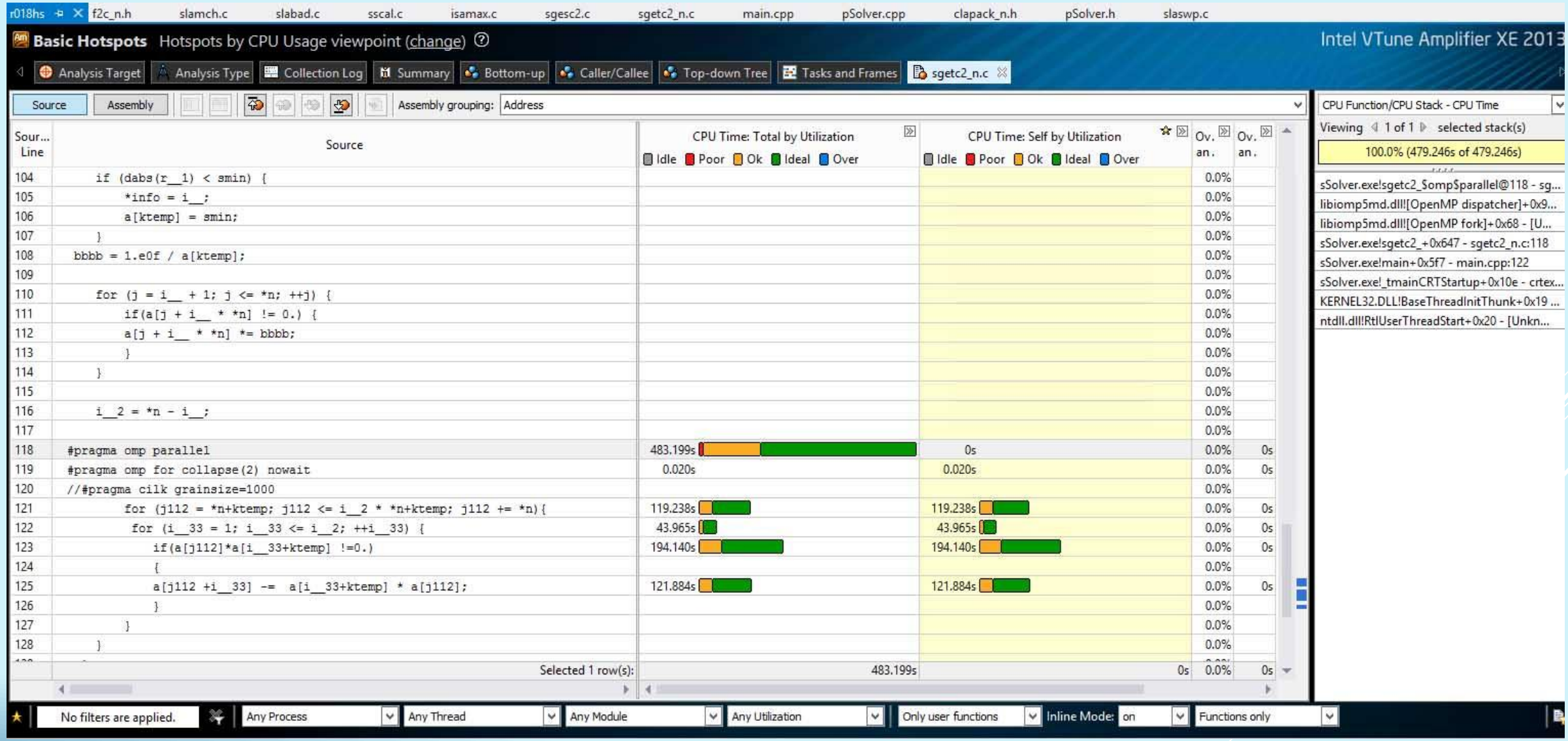
CPU Usage Histogram

This histogram represents a breakdown of the Elapsed Time. It visualizes what percentage of the wall time the specific number of CPUs were running simultaneously. CPU Usage may be higher than the thread concurrency if a thread is executing code on a CPU while it is logically waiting.



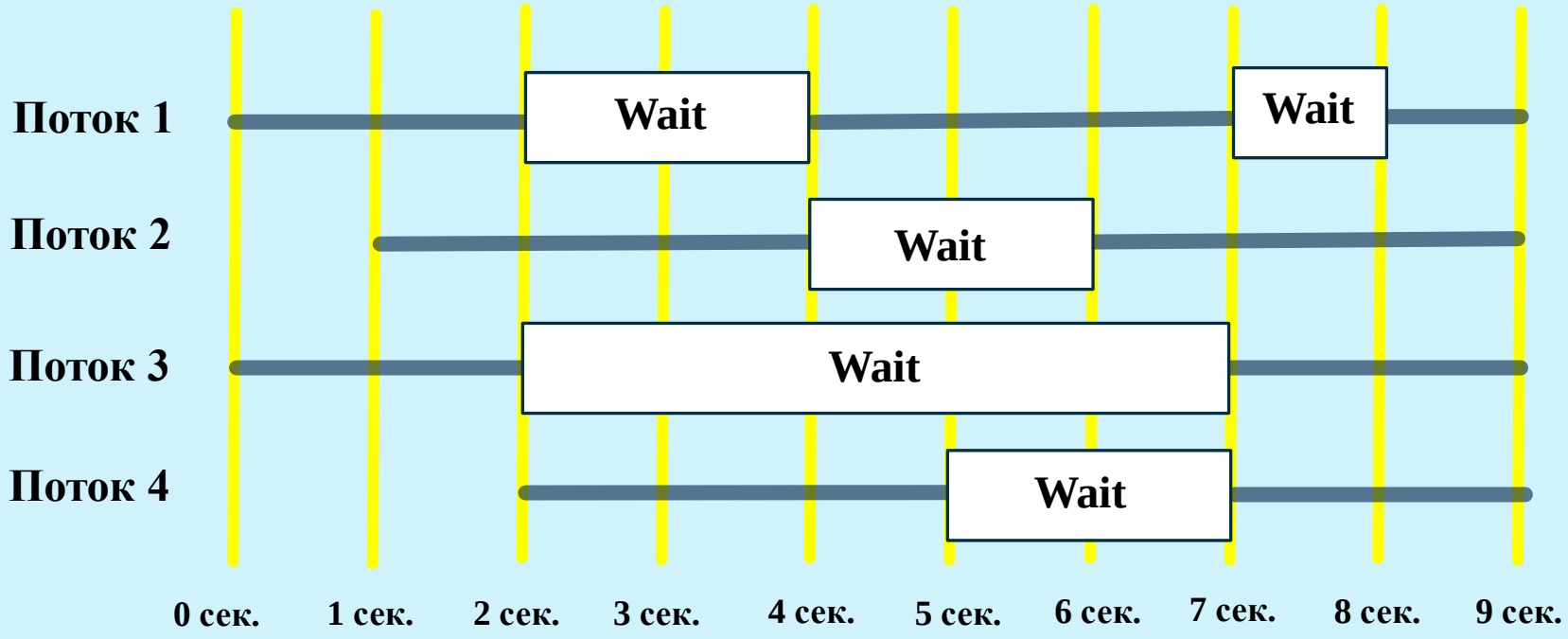
Вывод


r018hs
f2c_n.h
slamch.c
slabad.c
sscal.c
isamax.c
sgesc2.c
sgetc2_n.c
main.cpp
pSolver.cpp
clapack_n.h
pSolver.h
slaswp.c
Basic Hotspots
Hotspots by CPU Usage viewpoint (change)
Intel VTune Amplifier XE 2013
Analysis Target
Analysis Type
Collection Log
Summary
Bottom-up
Caller/Callee
Top-down Tree
Tasks and Frames
Grouping: Function / Call Stack
CPU Function/CPU Stack - CPU Time
Viewing 1 of 1 selected stack(s)
100.0% (479.246s of 479.246s)
Function / Call Stack
CPU Time by Utilization
Idle Poor Ok Ideal Over
Overhead and Spin Time
Module
Function (Full)
Source File
Start Address
sgetc2_ompparallel@118 479.246s 0s sSolver.exe sgetc2_ompparallel@118 sgetc2_n.c 0x1400043d8
sgetc2_ompparallel@40 291.179s 0s sSolver.exe sgetc2_ompparallel@40 sgetc2_n.c 0x140004537
[OpenMP worker] 111.837s 111.837s libiomp5md.dll __kmp_launch_worker(void *) 0x18009be80
[OpenMP fork] 5.803s 5.803s libiomp5md.dll _kmpc_fork_call 0x18004d080
main 3.356s 0s sSolver.exe main main.cpp 0x140001000
sgetc2_ 1.319s 0s sSolver.exe sgetc2_ sgetc2_n.c 0x140003c00
[OpenMP dispatcher] 0.594s 0.594s libiomp5md.dll _kmp_invoke_task_func 0x180070020
sgesc2_ 0.244s 0s sSolver.exe sgesc2_ sgesc2.c 0x140005960
_kmp_print_storage_map_gtid 0.207s 0.207s libiomp5md.dll _kmp_print_storage_map_gtid 0x180073a50
[Import thunk _kmpc_for_static_fini] 0.040s 0s sSolver.exe [Import thunk _kmpc_for_static_fini] 0x1400085ac
func@0x180002f2a 0.030s 0.030s ntdll.dll func@0x180002f2a 0x180002f2a
_kmp_x86_pause 0.030s 0.030s libiomp5md.dll _kmp_x86_pause 0x18009ad50
L_sgetc2__118__par_region2_2.2518 0.026s 0s sSolver.exe L_sgetc2__118__par_region2_2.2518 0x1400043d8
Selected 1 row(s): 479.246s 0s
sSolver.exe!sgetc2_ompparallel@118 - sg...
libiomp5md.dll![OpenMP dispatcher]+0x9...
libiomp5md.dll![OpenMP fork]+0x68 - [U...
sSolver.exe!sgetc2_+0x647 - sgetc2_n.c:118
sSolver.exe!main+0x5f7 - main.cpp:122
sSolver.exe!_tmainCRTStartup+0x10e - crtex...
KERNEL32.DLL!BaseThreadInitThunk+0x19 ...
ntdll.dll!RtlUserThreadStart+0x20 - [Unkn...
5s 10s 15s 20s 25s 30s 35s 40s 45s 50s 55s 60s 65s 70s 75s 80s 85s 90s 95s 100s 105s 110s 115s 120s 125s 130s 135s 140s
Thread
mainCRTStartup (0xe4c
OMP Worker Thread #9
OMP Worker Thread #6
OMP Worker Thread #1
OMP Worker Thread #1
OMP Worker Thread #4
OMP Worker Thread #5
OMP Worker Thread #1
OMP Worker Thread #8
CPU Usage
Thread
Running
CPU Time
Overhead an..
CPU Usage
CPU Time
Overhead an..
No filters are applied.
Any Process
Any Thread
Any Module
Any Utilization
Only user functions
Inline Mode: on
Functions only


r018hs
f2c_n.h
slamch.c
slabad.c
sscal.c
isamax.c
sgesc2.c
sgetc2_n.c
main.cpp
pSolver.cpp
clapack_n.h
pSolver.h
slaswp.c
Basic Hotspots
Hotspots by CPU Usage viewpoint (change)
Intel VTune Amplifier XE 2013
Analysis Target
Analysis Type
Collection Log
Summary
Bottom-up
Caller/Callee
Top-down Tree
Tasks and Frames
sgetc2_n.c
Source
Assembly
Assembly grouping:
Address
Sour... Line
Source
CPU Time: Total by Utilization
CPU Time: Self by Utilization
Idle Poor Ok Ideal Over
Ov. an.
104 if (dabs(r__1) < smin) {
105 *info = i__;
106 a[ktemp] = smin;
107 }
108 bbbb = 1.e0f / a[ktemp];
109
110 for (j = i__ + 1; j <= *n; ++j) {
111 if(a[j + i__ * *n] != 0.) {
112 a[j + i__ * *n] *= bbbb;
113 }
114 }
115
116 i__2 = *n - i__;
117
118 #pragma omp parallel 483.199s 0s 0.0% 0s
119 #pragma omp for collapse(2) nowait 0.020s 0.020s 0.0% 0s
120 //#pragma cilk grainsize=1000
121 for (j112 = *n+ktemp; j112 <= i__2 * *n+ktemp; j112 += *n){ 119.238s 119.238s 0.0% 0s
122 for (i__33 = 1; i__33 <= i__2; ++i__33) { 43.965s 43.965s 0.0% 0s
123 if(a[j112]*a[i__33+ktemp] !=0.) 194.140s 194.140s 0.0% 0s
124 {
125 a[j112 +i__33] -= a[i__33+ktemp] * a[j112]; 121.884s 121.884s 0.0% 0s
126 }
127 }
128 }
Selected 1 row(s): 483.199s 0s 0.0% 0s
CPU Function/CPU Stack - CPU Time
Viewing 1 of 1 selected stack(s)
100.0% (479.246s of 479.246s)
sSolver.exe!sgetc2_ompparallel@118 - sg...
libiomp5md.dll![OpenMP dispatcher]+0x9...
libiomp5md.dll![OpenMP fork]+0x68 - [U...
sSolver.exe!sgetc2_+0x647 - sgetc2_n.c:118
sSolver.exe!main+0x5f7 - main.cpp:122
sSolver.exe!_tmainCRTStartup+0x10e - crtex...
KERNEL32.DLL!BaseThreadInitThunk+0x19 ...
ntdll.dll!RtlUserThreadStart+0x20 - [Unkn...
No filters are applied.
Any Process
Any Thread
Any Module
Any Utilization
Only user functions
Inline Mode:
on
Functions only

Метрики производительности

Метрика	Описание
Available CPU Time	Общее время выполнения на всех ядрах
CPI	Cycles Per Instruction. Метрика, на которую следует обратить внимание в первую очередь. Хорошее значение 0.75, плохое значение 4. «Плохое» значение CPI может означать как неудачную организацию вычислений, так и неоптимальное использование логических ядер процессора (с технологией HT).
CPU Time	Время, затраченное активным процессором на выполнение потока. В многопоточном случае суммируется по всем потокам (логическим ядрам).
CPU Time by Utilization	Время, затраченное активным процессором на выполнение функции (время выполнения вызываемых функций не учитывается).
CPU Time:Total	Время, затраченное активным процессором на выполнение функции и всех вызываемых из неё функций.
Cx Residency	Время, проведённое в состоянии сна.
Elapsed Time	Время выполнения программы = время завершения программы – время запуска программы.



Elapsed Time = 9 секунд.

CPU Time = (2 + 3 + 1) + (3 + 3) + (2 + 2) + (3 + 2) = 21 секунда.

Wait Time = 2 + 1 + 5 + 2 = 10 секунд.

Метрика	Описание
Idle Time	Время, в течение которого нить неактивна, а система не может переключиться на выполнение другой задачи.
Inactive Time	Время, в течение которого нить остаётся вытесненной из исполнения.
Overhead Time	Время, в течение которого проходит с момента освобождения ресурса его предыдущим обладателем до захвата следующим собственником. Чем меньше это время, тем лучше. Если объём работы фиксирован, а степень параллелизма увеличивается, Overhead Time будет расти.
Spin Time	Wait Time, в течение которого процессор был занят. Spin возникает, когда программные средства синхронизации выполняют опрос ЦП, в то время как поток находится в состоянии ожидания.
Total Thread Count	Количество исполняющихся потоков.
Wait Time	Длительность простоя потока в результате действия механизмов синхронизации.
Synchronization Context Switches	Количество переключений контекста в результате явного вызова функций синхронизации.
и другие, всего около 35	

Метрики производительности, основанные на аппаратных событиях

Метрика	Событие	Описание
False Sharing		Простой процессора в результате попытки доступа к разным элементам данных, находящихся в одной строке кэш-памяти.
Assists	`FP_ASSIST.ANY`	Некоторые операции могут быть выполнены не аппаратно, а только с помощью микрокода (например, арифметические операции с денормализованными числами). Объём микропрограммы – несколько сотен команд.
L1 Misses		Промахи кэш-памяти 1 уровня. Для Intel® Xeon™ Phi.
Branch Mispredict	`BR_MISP_RETIRED.ALL_BRANCHES_PS`	Ошибка предсказания ветвления. Если ветвление предсказано неправильно, инструкции из неправильно предсказанной ветви всё равно выполняются. Это лишние затраты времени.
Cache Usage		Эффективность использования кэш-памяти. Связана с локальностью данных. Характерное время доступа к данным из L1 20 тактов, из L2 250 тактов.
LLC Miss		Промахи кэш-памяти уровня, ближайшего к оперативной памяти.
Split Loads/Split Stores		Загрузка/выгрузка элемента данных, когда он занимает 2 строки кэш-памяти (длина строки 64 байта).

Метрика	Событие	Описание
DTLB Overhead	`DTLB_LOAD_MISSES.STLB_HIT` `DTLB_MISSES`	DTLB (Data Translation Look-ahead Buffers) позволяют минимизировать обращения к таблице страниц, используемой при работе с виртуальной памятью. Буферы организованы в многоуровневую иерархию, чем больше путь к требуемым данным, тем больше накладные расходы.
Memory Bus Transactions		Количество транзакций через шину.
Vectorization Usage		Для Intel® Xeon™ Phi. Отношение числа элементов данных, обработанных векторными инструкциями, к числу векторных инструкций. Значения 8, 16 соответствуют хорошей векторизации циклов.
Bus Lock		Блокировки доступа к шине. Возникают при использовании различных видов синхронизации.
Execution Stalls	`RESOURCE_STALLS.ANY`	Вычислительные ресурсы ЦП используются полностью, однако операции с большой латентностью сериализуются, ожидая критически важные ресурсы.
Issued uOps	`UOPS_ISSUED.ANY`	Количество микроинструкций, выполняемых ядром. На каждом такте аппаратный поток генерирует до 4 микроинструкций. Если реальное количество меньше, производительность.
ICache Misses	`ICACHE.MISSES`	Промахи кэш-памяти декодированных инструкций.
Другие.		

Intel® VTune™ Amplifier позволяет:

- определить, какие части программы содержат «хотспоты» (потребляют большую часть процессорного времени);
- определить, какие последовательности вызовов определяют производительность приложения;
- определить, какие части программы неэффективно используют процессор;
- выявить объекты синхронизации, которые негативно влияют на производительность программы;
- локализовать участки кода, в которых неэффективно используются аппаратные ресурсы: память, шина и др.;
- анализировать производительность, конфликты потоков, простои, энергопотребление;
- определить, где и почему работа приложения приводит к неэффективному потреблению электроэнергии (только в Linux);
- другие виды анализа (в том числе Frame и Task анализ).

Результаты расширенного анализа аппаратной EBS-статистики отображаются с помощью предустановленных средств просмотра:

- **Hotspots** – определение фрагментов программы, на выполнение которых затрачивается наибольшая часть процессорного времени.
- **Hardware Event Counts** – счётчик событий процессора.
- **Hardware Event Sample Counts** – счётчик полной статистики событий процессора.
- **Hardware Issues** – определение фрагментов программы, неэффективно использующих аппаратные ресурсы.
- **General Exploration** – определение проблем, связанных с неэффективным использованием ресурсов микроархитектуры Sandy Bridge и Intel® Atom™.
- **Bandwidth** – определение фрагментов программы, генерирующих большой поток данных в DRAM для микроархитектур Sandy Bridge, Ivy Bridge и Nehalem.
- Если выбрана опция **Collect stacks**, все средства просмотра дополняются метриками, помогающими определить критические пути вызовов функций, проанализировать конфликты потоков и энергопотребление для этих путей.

Эффективность использования процессора

- ***Idle*** – все ядра находятся в состоянии ожидания. Ни один поток не выполняется.
- ***Poor*** – доля одновременно работающих ядер < 50%.
- ***Ok*** – доля одновременно работающих ядер от 51% до 85%.
- ***Ideal*** – доля одновременно работающих ядер > 86%.

Эффективность многопоточного параллелизма

- ***Idle*** – все потоки находятся в состоянии ожидания. Ни один поток не выполняется.
- ***Poor*** – низкая эффективность. Доля используемых потоков < 50% от степени параллелизма архитектуры.
- ***Ok*** – хорошая эффективность. Доля используемых потоков от 51% до 85% от степени параллелизма архитектуры.
- ***Ideal*** – очень хорошая эффективность. Доля используемых потоков от 86% до 115% от степени параллелизма архитектуры.
- ***Over*** – избыточная эффективность. Доля используемых потоков > 115% от степени параллелизма архитектуры.

Виды анализа Intel® Vtune Amplifier XE

Algorithm analysis

- **Basic Hotspots.** Интервал сбора статистики 10 мс. Используются ЦП таймеры высокой точности. Интервал сбора статистики GPU 1000 мкс. Построение временной диаграммы. Сбор статистики по пользовательским задачам.
- **Advanced Hotspots.** Используется для определения положения «хотспотов». В дополнение к базовому уровню анализируются стеки вызовов, переключения контекста, анализируется метрика CPI и другие аспекты. Анонсированы небольшие накладные расходы. Для сбора статистики используется драйвер. Статистика собирается по событиям CPU_CLK_UNHALTED.REF_TSC, CPU_CLK_UNHALTED.THREAD, INST_RETIRED.ANY.
- **Concurrency.** Анализ реализации многопоточного параллелизма на уровне логических ядер. Этот вид анализа может использоваться для поиска избыточной синхронизации и накладных расходов.
- **Locks and Waits.** Блокировки и простои. Этот вид анализа позволяет определить ожжидание при синхронизации, операций ввода-вывода, а также их влияние на производительность приложения.

Intel Core 2 Processor Analysis

- General Exploration.
- Memory Accsess.
- Bandwidth.
- Bandwidth Breakdown.
- Cycles and uOps.

Nehalem / Westmere Analysis
(Core i5, Core i7, Xeon E7, …)

- General Exploration.
- Memory Access.
- Read Bandwidth.
- Write Bandwidth.
- Cycles and uOps.
- Front End Investigation.

Sandy Bridge / Ivy Bridge / Haswell Analysis
(Intel® Xeon™ E5, Core i3, Core i5, Core i7, …)

- **General Exploration.** Около 40 событий.
- **Memory Access**. Более 10 событий.
- **Bandwidth**. Оценка объёма данных, считываемых из и записываемых в память через контроллер памяти, определение, достигается ли насыщение пропускной способности. 4 события.
- **Access Contention**. Анализ работы с кэш-памятью, блокировок, влияющих на быстродействие и т.л.
- **Cycles and uOps**.
- **Branch Analysis**. Анализ эффективности предсказания ветвлений.
- **Client Analysis**.
- **Core Port Saturation**.
- **Port Saturation**.

Intel Atom Processor Analysis
General Exploration.

Knights Corner Platform Analysis
(Intel® Xeon™ Phi)
- Hotspots.
- General Exploration.
- Bandwidth.

Power Analysis
- CPU Sleep States.
- CPU Frequency.

Custom Analysis

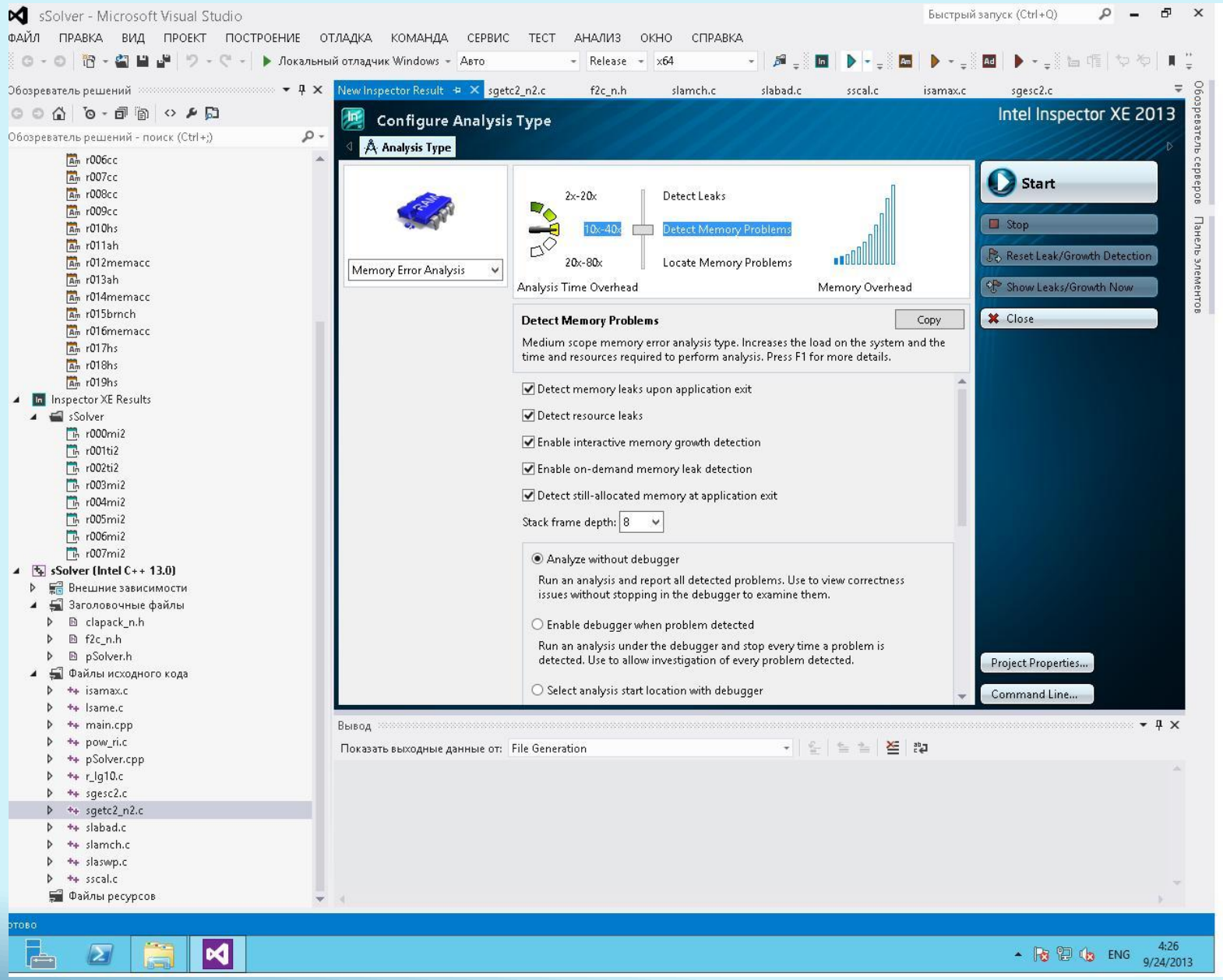
Intel® Inspector XE 2013

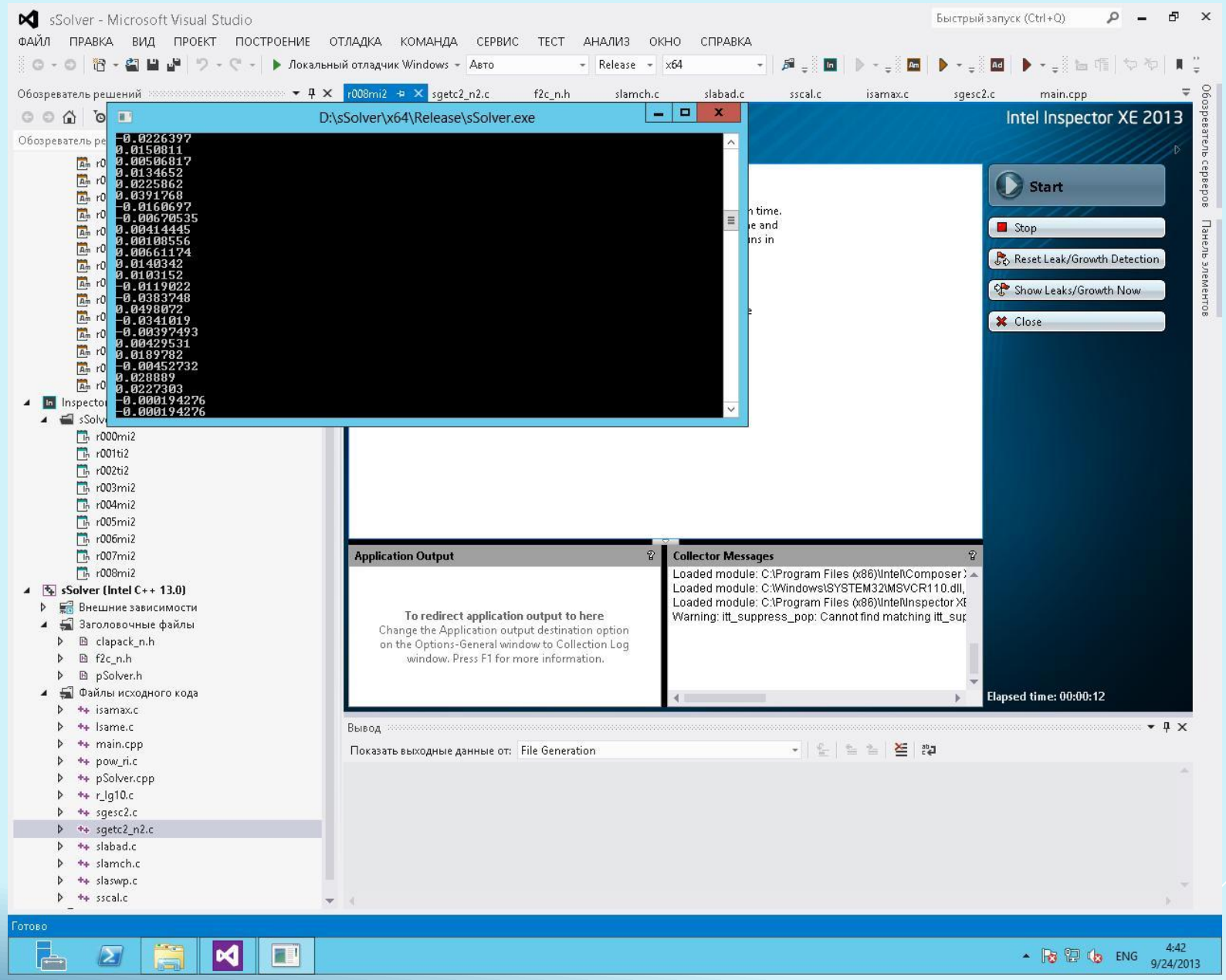
Многоплатформенный: MS Windows, Linux.
Поддержка: C/C++, C#, Fortran.
Интеграция с Microsoft Visual Studio (включая VS 2012).
Специальная подготовка кода не требуется.

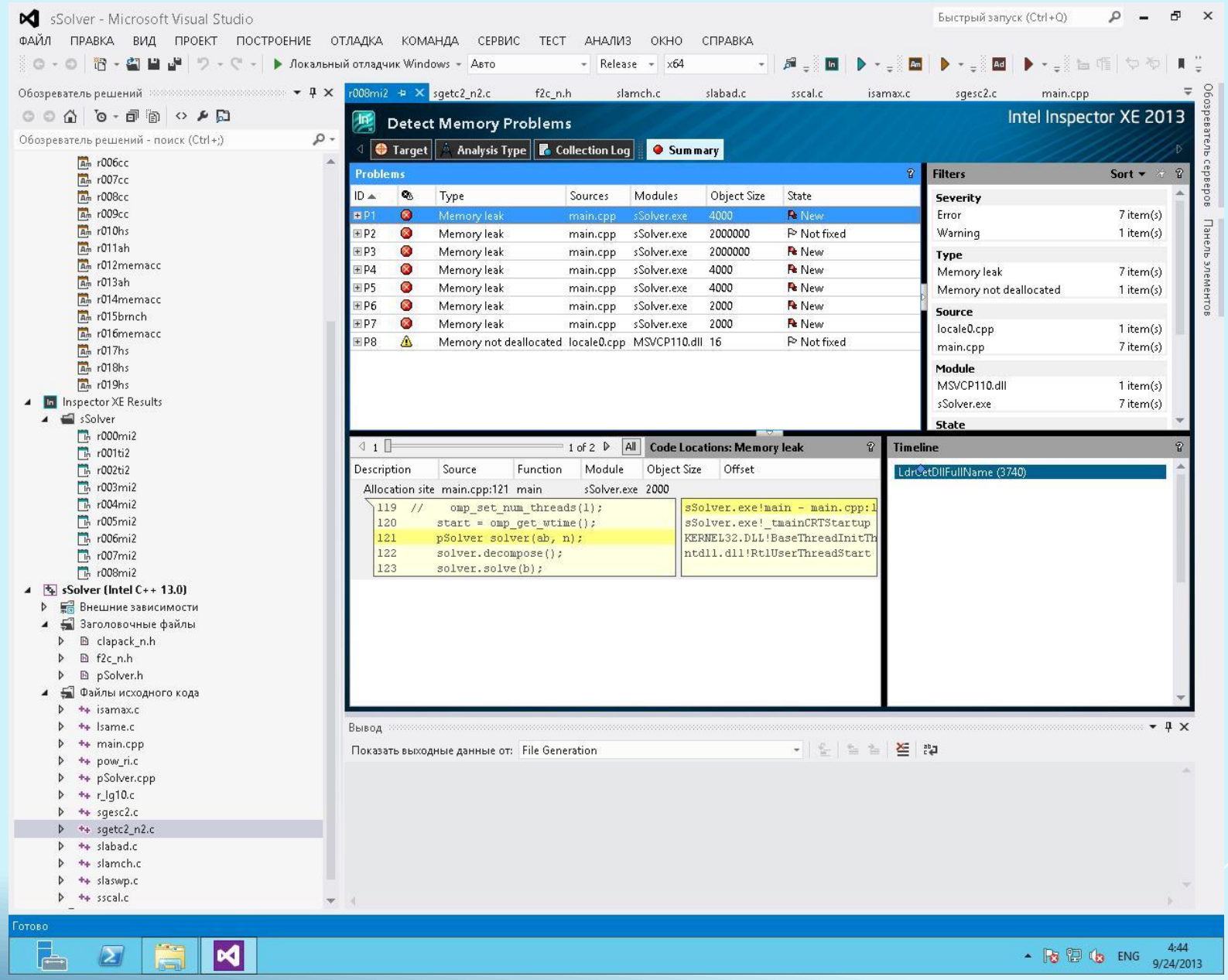
Поиск и локализация:

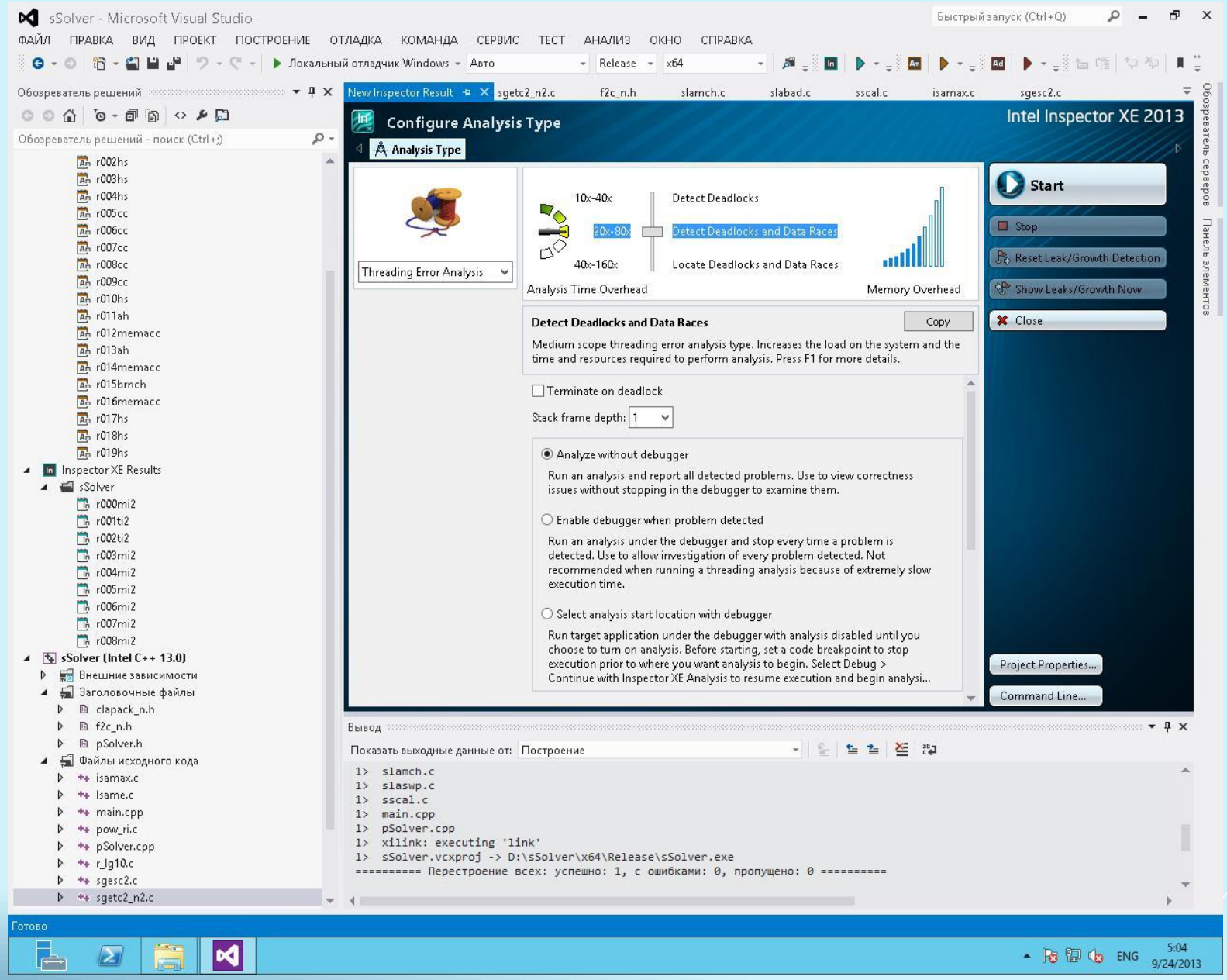
- некорректного доступа к памяти;
- утечек памяти;
- некорректного использования указателей;
- переполнения буферов;
- гонок за данными (heap races, stack races);
- взаимных блокировок;
- доступа к чужим стекам;
- и других ошибок работы с памятью и реализации многопоточности.

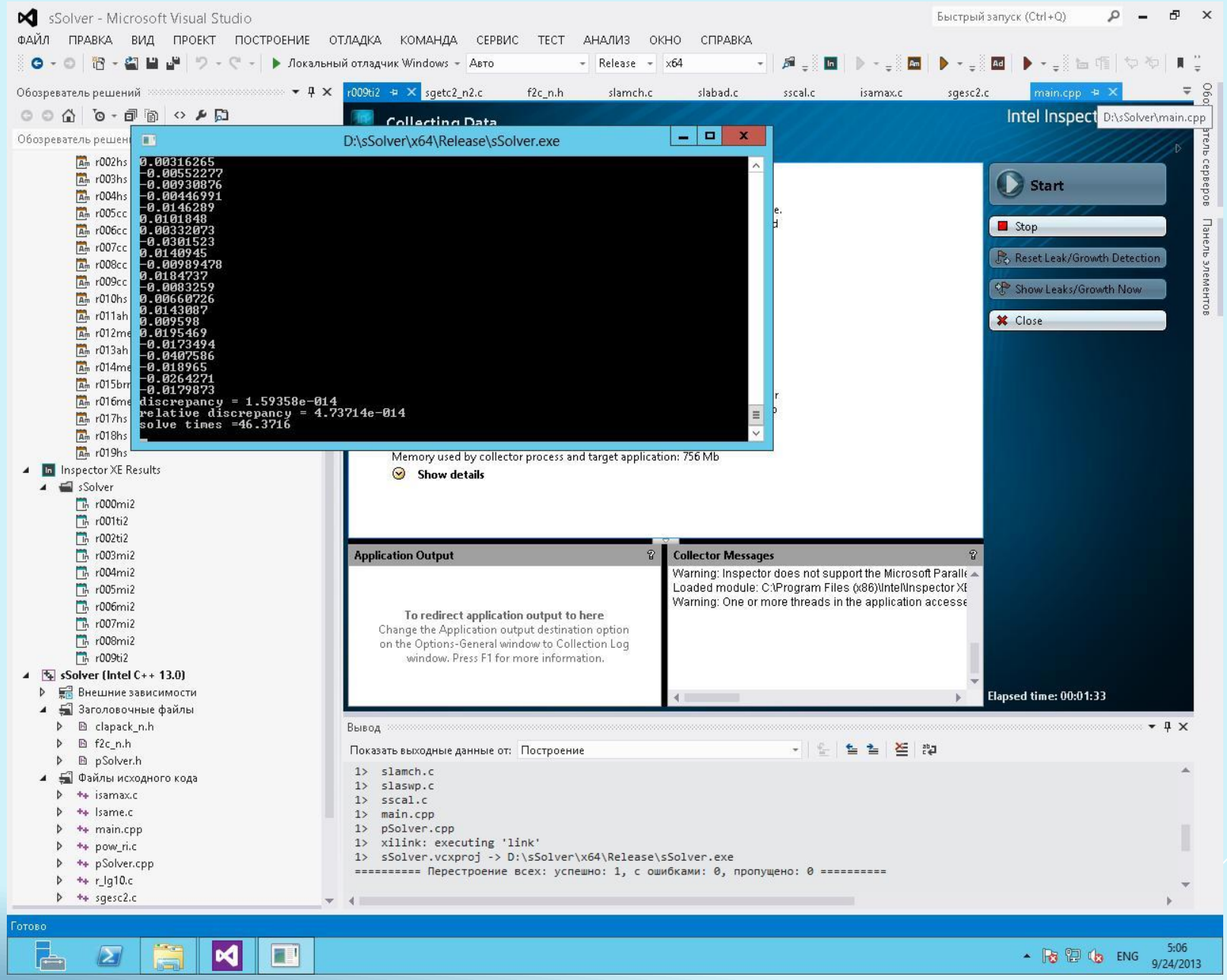
Поддержка Intel® Xeon™ Phi.
Возможность анализа MPI-приложений на ошибки памяти и ошибки многопоточности.
Поддержка Intel® TBB, Intel® Cilk™ Plus, OpenMP.
Возможность ограничивать область анализа.
Поддержка командной работы над проектом.

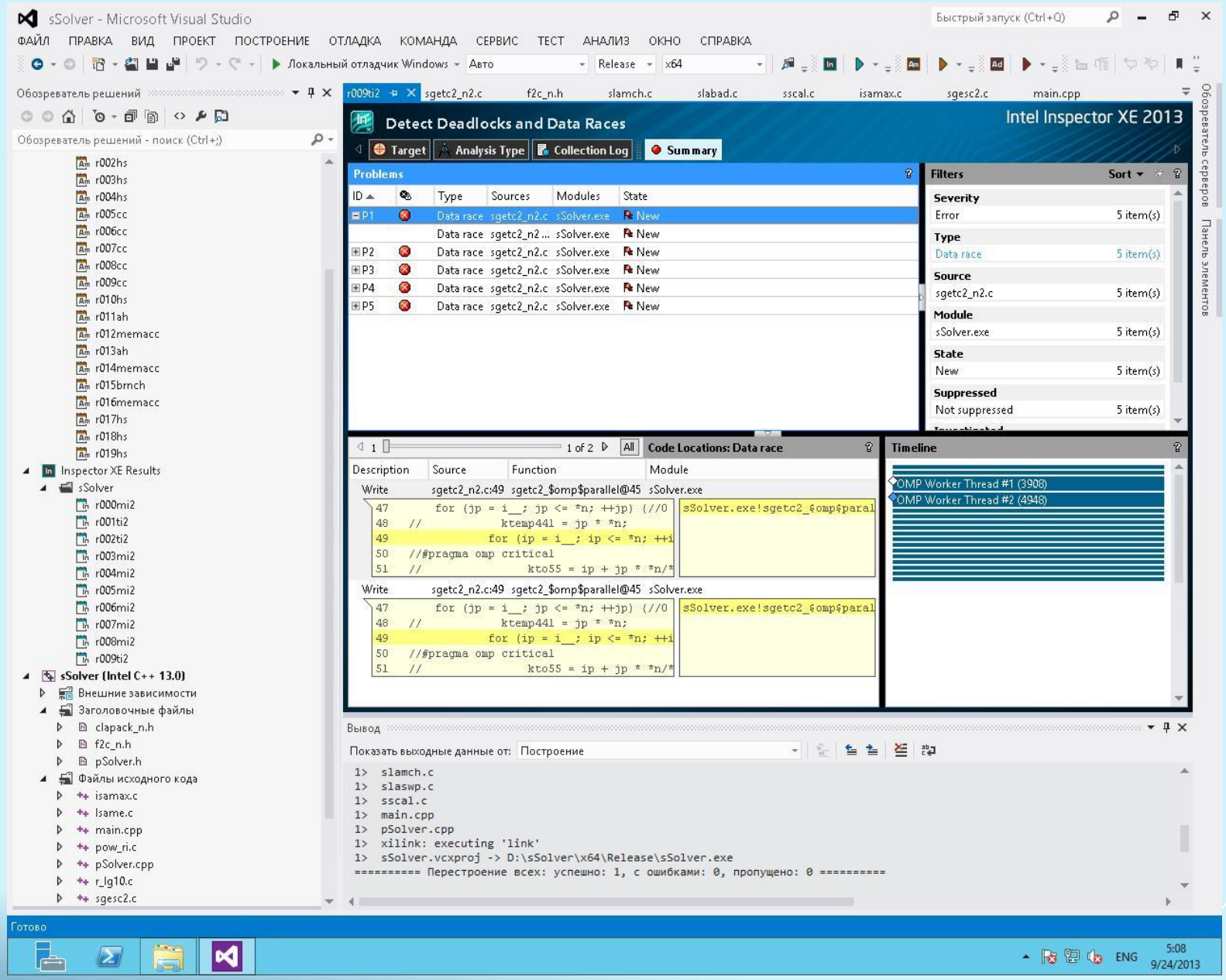

sSolver - Microsoft Visual Studio
ФАЙЛ ПРАВКА ВИД ПРОЕКТ ПОСТРОЕНИЕ ОТЛАДКА КОМАНДА СЕРВИС ТЕСТ АНАЛИЗ ОКНО СПРАВКА
Обозреватель решений
Inspector XE Results
sSolver
sSolver (Intel C++ 13.0)
Внешние зависимости
Заголовочные файлы
Файлы исходного кода
Файлы ресурсов
New Inspector Result
Configure Analysis Type
Intel Inspector XE 2013
Analysis Type
Memory Error Analysis
2x-20x
10x-40x
20x-80x
Detect Leaks
Detect Memory Problems
Locate Memory Problems
Analysis Time Overhead
Memory Overhead
Start
Stop
Reset Leak/Growth Detection
Show Leaks/Growth Now
Close
Detect Memory Problems
Copy
Medium scope memory error analysis type. Increases the load on the system and the time and resources required to perform analysis. Press F1 for more details.
Detect memory leaks upon application exit
Detect resource leaks
Enable interactive memory growth detection
Enable on-demand memory leak detection
Detect still-allocated memory at application exit
Stack frame depth: 8
Analyze without debugger
Run an analysis and report all detected problems. Use to view correctness issues without stopping in the debugger to examine them.
Enable debugger when problem detected
Run an analysis under the debugger and stop every time a problem is detected. Use to allow investigation of every problem detected.
Select analysis start location with debugger
Project Properties...
Command Line...
Вывод
Показать выходные данные от: File Generation
ENG
4:26
9/24/2013

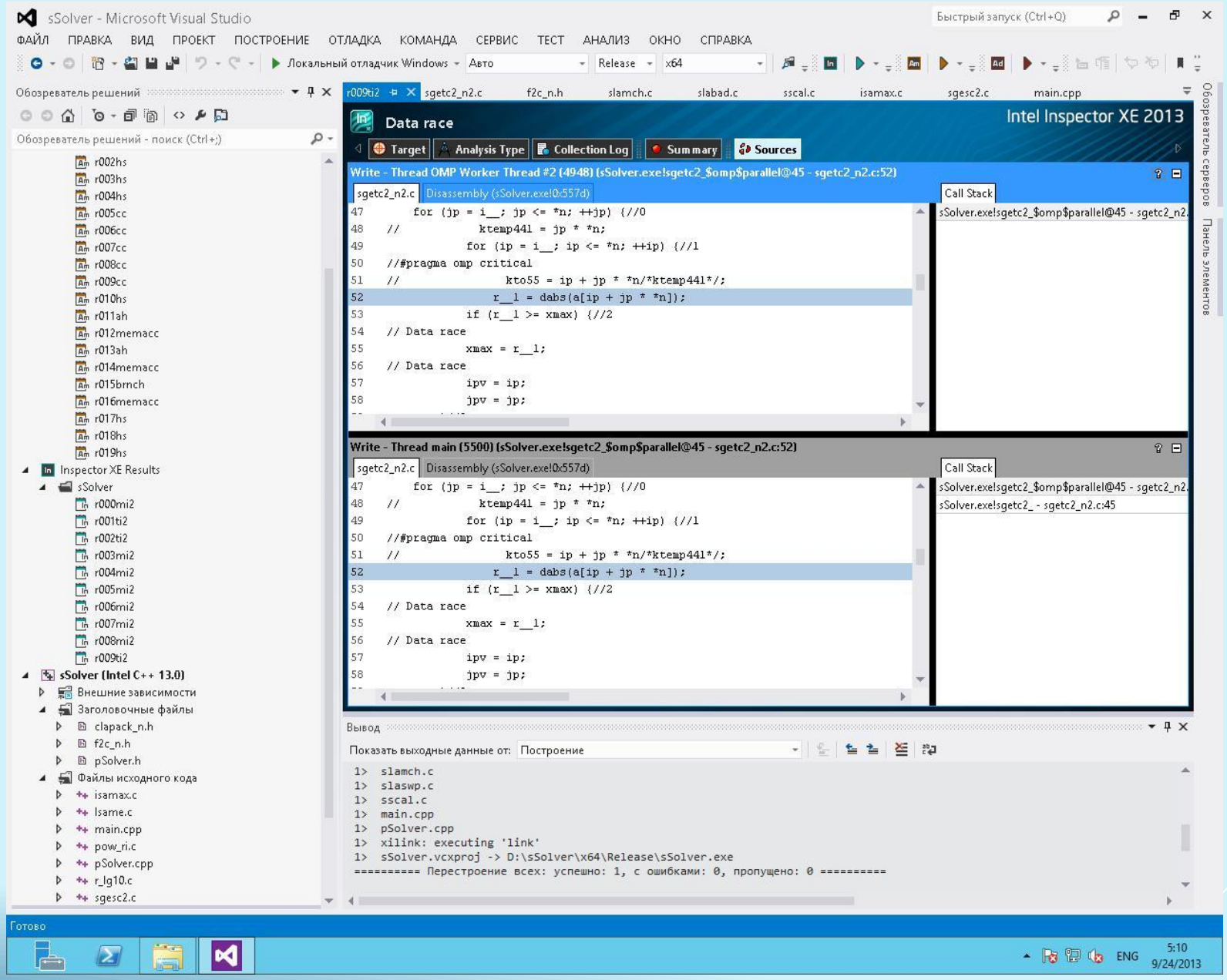

sSolver - Microsoft Visual Studio
D:\sSolver\x64\Release\sSolver.exe
-0.0226397
0.0150811
0.00506817
0.0134652
0.0225862
0.0391768
-0.0160697
-0.00670535
0.00414445
0.00108556
0.00661174
0.0140342
0.0103152
-0.0119022
-0.0383748
0.0498072
-0.0341019
-0.00397493
0.00429531
0.0189782
-0.00452732
0.028889
0.0227303
-0.000194276
-0.000194276
Intel Inspector XE 2013
Start
Stop
Reset Leak/Growth Detection
Show Leaks/Growth Now
Close
Application Output
To redirect application output to here
Change the Application output destination option on the Options-General window to Collection Log window. Press F1 for more information.
Collector Messages
Loaded module: C:\Program Files (x86)\Intel\Composer
Loaded module: C:\Windows\SYSTEM32\MSVCR110.dll,
Loaded module: C:\Program Files (x86)\Intel\Inspector XE
Warning: itt_suppress_pop: Cannot find matching itt_sup
Elapsed time: 00:00:12
Вывод
Показать выходные данные от: File Generation
Готово
ENG
4:42
9/24/2013

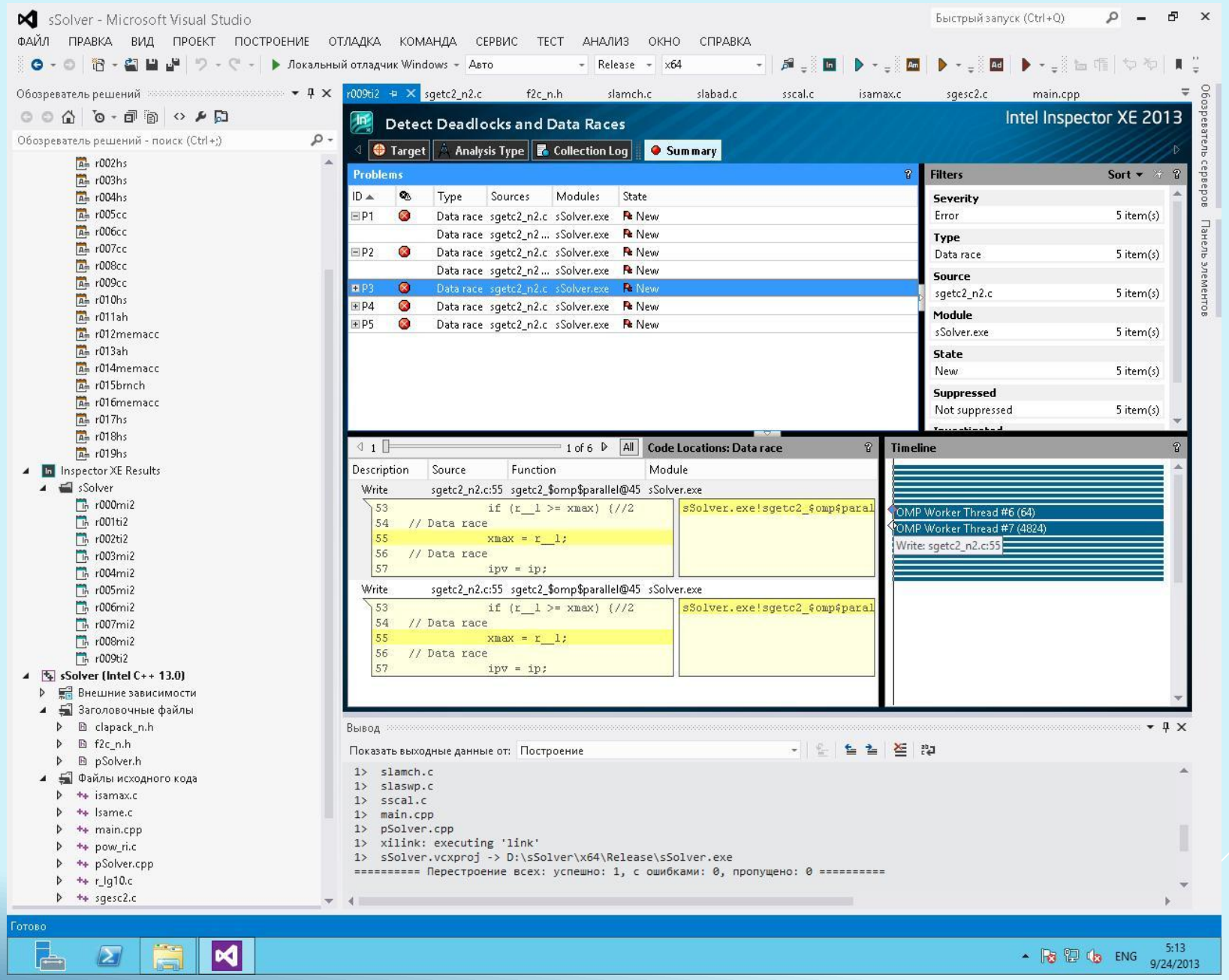

sSolver - Microsoft Visual Studio
Быстрый запуск (Ctrl+Q)
ФАЙЛ ПРАВКА ВИД ПРОЕКТ ПОСТРОЕНИЕ ОТЛАДКА КОМАНДА СЕРВИС ТЕСТ АНАЛИЗ ОКНО СПРАВКА
Локальный отладчик Windows
Авто
Release
x64
Обозреватель решений
Обозреватель решений - поиск (Ctrl+;)
r006cc
r007cc
r008cc
r009cc
r010hs
r011ah
r012memacc
r013ah
r014memacc
r015brnch
r016memacc
r017hs
r018hs
r019hs
Inspector XE Results
sSolver
r000mi2
r001ti2
r002ti2
r003mi2
r004mi2
r005mi2
r006mi2
r007mi2
r008mi2
sSolver (Intel C++ 13.0)
Внешние зависимости
Заголовочные файлы
clapack_n.h
f2c_n.h
pSolver.h
Файлы исходного кода
isamax.c
lsame.c
main.cpp
pow_ri.c
pSolver.cpp
r_lg10.c
sgesc2.c
sgetc2_n2.c
slabad.c
slamch.c
slaswp.c
sscal.c
r008mi2
sgetc2_n2.c
f2c_n.h
slamch.c
slabad.c
sscal.c
isamax.c
sgesc2.c
main.cpp
Detect Memory Problems
Intel Inspector XE 2013
Target
Analysis Type
Collection Log
Summary
Problems
ID
Type
Sources
Modules
Object Size
State
P1 Memory leak main.cpp sSolver.exe 4000 New
P2 Memory leak main.cpp sSolver.exe 2000000 Not fixed
P3 Memory leak main.cpp sSolver.exe 2000000 New
P4 Memory leak main.cpp sSolver.exe 4000 New
P5 Memory leak main.cpp sSolver.exe 4000 New
P6 Memory leak main.cpp sSolver.exe 2000 New
P7 Memory leak main.cpp sSolver.exe 2000 New
P8 Memory not deallocated locale0.cpp MSVCP110.dll 16 Not fixed
Filters
Sort
Severity
Error 7 item(s)
Warning 1 item(s)
Type
Memory leak 7 item(s)
Memory not deallocated 1 item(s)
Source
locale0.cpp 1 item(s)
main.cpp 7 item(s)
Module
MSVCP110.dll 1 item(s)
sSolver.exe 7 item(s)
State
1 of 2
All
Code Locations: Memory leak
Description
Source
Function
Module
Object Size
Offset
Allocation site main.cpp:121 main sSolver.exe 2000
119 // omp_set_num_threads(1);
120 start = omp_get_wtime();
121 pSolver solver(ab, n);
122 solver.decompose();
123 solver.solve(b);
sSolver.exe!main - main.cpp:1
sSolver.exe!_tmainCRTStartup
KERNEL32.DLL!BaseThreadInitTh
ntdll.dll!RtlUserThreadStart
Timeline
LdrGetDllFullName (3740)
Вывод
Показать выходные данные от: File Generation
Готово
ENG
4:44
9/24/2013

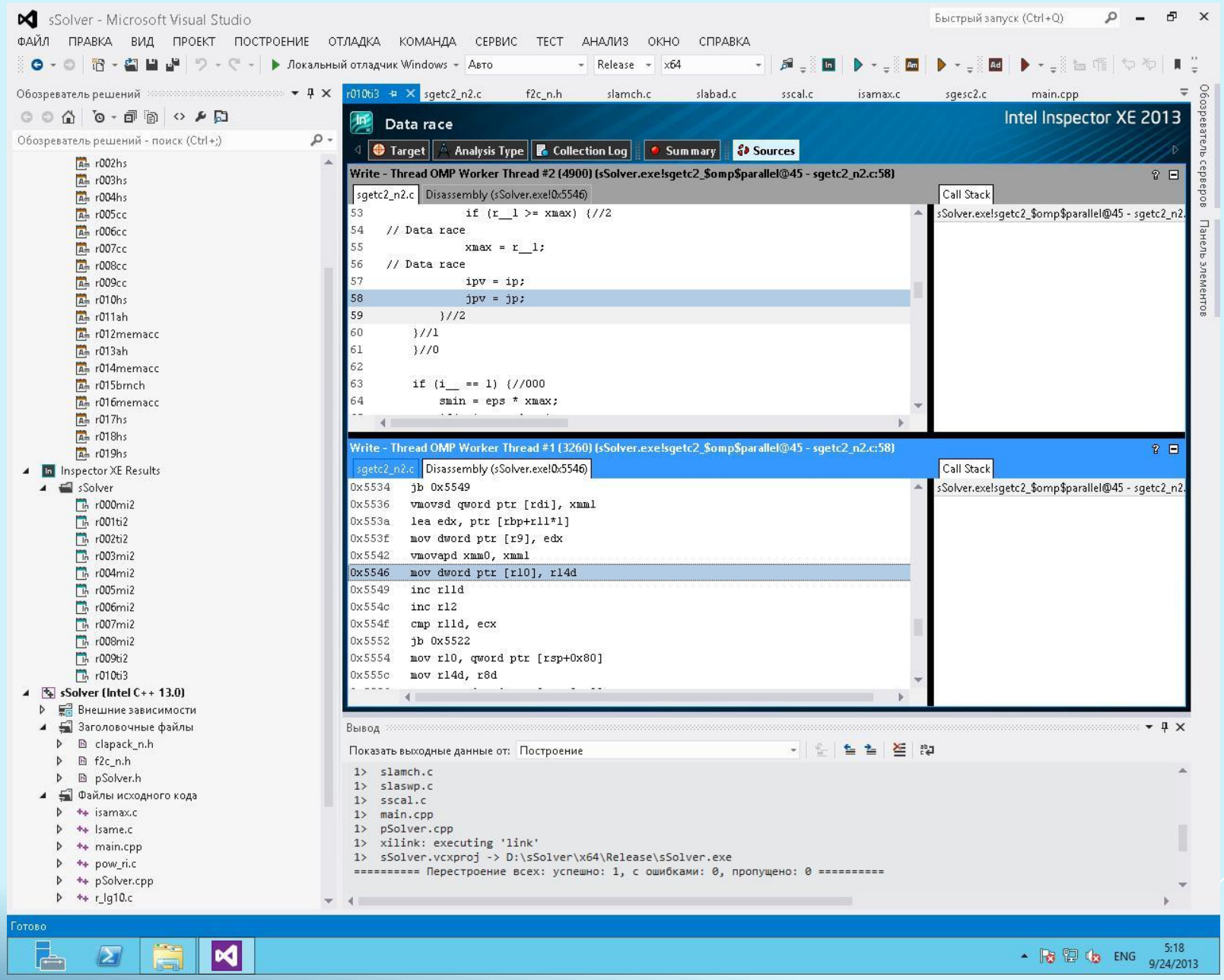

sSolver - Microsoft Visual Studio
Configure Analysis Type
Intel Inspector XE 2013
Analysis Type
Start
Stop
Reset Leak/Growth Detection
Show Leaks/Growth Now
Close
Detect Deadlocks
Detect Deadlocks and Data Races
Locate Deadlocks and Data Races
Threading Error Analysis
Copy
Terminate on deadlock
Stack frame depth:
Analyze without debugger
Enable debugger when problem detected
Select analysis start location with debugger
Project Properties...
Command Line...
Вывод
Готово


sSolver - Microsoft Visual Studio
D:\sSolver\x64\Release\sSolver.exe
0.00316265
-0.00552277
-0.00930876
-0.00446991
-0.0146289
0.0101848
0.00332073
-0.0301523
0.0140945
-0.00989478
0.0184737
-0.0083259
0.00660726
0.0143087
0.009598
0.0195469
-0.0173494
-0.0407586
-0.018965
-0.0264271
-0.0179873
discrepancy = 1.59358e-014
relative discrepancy = 4.73714e-014
solve times =46.3716
Start
Stop
Reset Leak/Growth Detection
Show Leaks/Growth Now
Close
Memory used by collector process and target application: 756 Mb
Show details
Application Output
To redirect application output to here
Collector Messages
Elapsed time: 00:01:33
1> slamch.c
1> slaswp.c
1> sscal.c
1> main.cpp
1> pSolver.cpp
1> xilink: executing 'link'
1> sSolver.vcxproj -> D:\sSolver\x64\Release\sSolver.exe
========== Перестроение всех: успешно: 1, с ошибками: 0, пропущено: 0 ==========
Готово


sSolver - Microsoft Visual Studio
Быстрый запуск (Ctrl+Q)
ФАЙЛ ПРАВКА ВИД ПРОЕКТ ПОСТРОЕНИЕ ОТЛАДКА КОМАНДА СЕРВИС ТЕСТ АНАЛИЗ ОКНО СПРАВКА
Локальный отладчик Windows
Авто
Release
x64
Обозреватель решений
Обозреватель решений - поиск (Ctrl+;)
r002hs
r003hs
r004hs
r005cc
r006cc
r007cc
r008cc
r009cc
r010hs
r011ah
r012memacc
r013ah
r014memacc
r015brnch
r016memacc
r017hs
r018hs
r019hs
Inspector XE Results
sSolver
r000mi2
r001ti2
r002ti2
r003mi2
r004mi2
r005mi2
r006mi2
r007mi2
r008mi2
r009ti2
sSolver (Intel C++ 13.0)
Внешние зависимости
Заголовочные файлы
clapack_n.h
f2c_n.h
pSolver.h
Файлы исходного кода
isamax.c
lsame.c
main.cpp
pow_ri.c
pSolver.cpp
r_lg10.c
sgesc2.c
r009ti2
sgetc2_n2.c
f2c_n.h
slamch.c
slabad.c
sscal.c
isamax.c
sgesc2.c
main.cpp
Detect Deadlocks and Data Races
Intel Inspector XE 2013
Target
Analysis Type
Collection Log
Summary
Problems
ID
Type
Sources
Modules
State
P1 Data race sgetc2_n2.c sSolver.exe New
Data race sgetc2_n2 ... sSolver.exe New
P2 Data race sgetc2_n2.c sSolver.exe New
P3 Data race sgetc2_n2.c sSolver.exe New
P4 Data race sgetc2_n2.c sSolver.exe New
P5 Data race sgetc2_n2.c sSolver.exe New
Filters
Sort
Severity
Error 5 item(s)
Type
Data race 5 item(s)
Source
sgetc2_n2.c 5 item(s)
Module
sSolver.exe 5 item(s)
State
New 5 item(s)
Suppressed
Not suppressed 5 item(s)
1 of 2
All
Code Locations: Data race
Description
Source
Function
Module
Write sgetc2_n2.c:49 sgetc2_ompparallel@45 sSolver.exe
47 for (jp = i__; jp <= *n; ++jp) {//0
48 // ktemp44l = jp * *n;
49 for (ip = i__; ip <= *n; ++i
50 //#pragma omp critical
51 // kto55 = ip + jp * *n/*
sSolver.exe!sgetc2_ompparal
Write sgetc2_n2.c:49 sgetc2_ompparallel@45 sSolver.exe
47 for (jp = i__; jp <= *n; ++jp) {//0
48 // ktemp44l = jp * *n;
49 for (ip = i__; ip <= *n; ++i
50 //#pragma omp critical
51 // kto55 = ip + jp * *n/*
sSolver.exe!sgetc2_ompparal
Timeline
OMP Worker Thread #1 (3908)
OMP Worker Thread #2 (4948)
Вывод
Показать выходные данные от: Построение
1> slamch.c
1> slaswp.c
1> sscal.c
1> main.cpp
1> pSolver.cpp
1> xilink: executing 'link'
1> sSolver.vcxproj -> D:\sSolver\x64\Release\sSolver.exe
========== Перестроение всех: успешно: 1, с ошибками: 0, пропущено: 0 ==========
Готово
ENG
5:08
9/24/2013


sSolver - Microsoft Visual Studio
Data race
Intel Inspector XE 2013
Target
Analysis Type
Collection Log
Summary
Sources
Write - Thread OMP Worker Thread #2 (4948) (sSolver.exe!sgetc2_ompparallel@45 - sgetc2_n2.c:52)
Write - Thread main (5500) (sSolver.exe!sgetc2_ompparallel@45 - sgetc2_n2.c:52)
Call Stack
Вывод
Готово


sSolver - Microsoft Visual Studio
Быстрый запуск (Ctrl+Q)
ФАЙЛ ПРАВКА ВИД ПРОЕКТ ПОСТРОЕНИЕ ОТЛАДКА КОМАНДА СЕРВИС ТЕСТ АНАЛИЗ ОКНО СПРАВКА
Локальный отладчик Windows
Авто
Release
x64
Обозреватель решений
Обозреватель решений - поиск (Ctrl+;)
r002hs
r003hs
r004hs
r005cc
r006cc
r007cc
r008cc
r009cc
r010hs
r011ah
r012memacc
r013ah
r014memacc
r015brnch
r016memacc
r017hs
r018hs
r019hs
Inspector XE Results
sSolver
r000mi2
r001ti2
r002ti2
r003mi2
r004mi2
r005mi2
r006mi2
r007mi2
r008mi2
r009ti2
sSolver (Intel C++ 13.0)
Внешние зависимости
Заголовочные файлы
clapack_n.h
f2c_n.h
pSolver.h
Файлы исходного кода
isamax.c
lsame.c
main.cpp
pow_ri.c
pSolver.cpp
r_lg10.c
sgesc2.c
r009ti2
sgetc2_n2.c
f2c_n.h
slamch.c
slabad.c
sscal.c
isamax.c
sgesc2.c
main.cpp
Detect Deadlocks and Data Races
Intel Inspector XE 2013
Target
Analysis Type
Collection Log
Summary
Problems
ID Type Sources Modules State
P1 Data race sgetc2_n2.c sSolver.exe New
Data race sgetc2_n2 ... sSolver.exe New
P2 Data race sgetc2_n2.c sSolver.exe New
Data race sgetc2_n2 ... sSolver.exe New
P3 Data race sgetc2_n2.c sSolver.exe New
P4 Data race sgetc2_n2.c sSolver.exe New
P5 Data race sgetc2_n2.c sSolver.exe New
Filters
Sort
Severity
Error 5 item(s)
Type
Data race 5 item(s)
Source
sgetc2_n2.c 5 item(s)
Module
sSolver.exe 5 item(s)
State
New 5 item(s)
Suppressed
Not suppressed 5 item(s)
1 of 6
All
Code Locations: Data race
Description Source Function Module
Write sgetc2_n2.c:55 sgetc2_ompparallel@45 sSolver.exe
53 if (r__1 >= xmax) {//2
54 // Data race
55 xmax = r__1;
56 // Data race
57 ipv = ip;
sSolver.exe!sgetc2_ompparal
Write sgetc2_n2.c:55 sgetc2_ompparallel@45 sSolver.exe
53 if (r__1 >= xmax) {//2
54 // Data race
55 xmax = r__1;
56 // Data race
57 ipv = ip;
sSolver.exe!sgetc2_ompparal
Timeline
OMP Worker Thread #6 (64)
OMP Worker Thread #7 (4824)
Write: sgetc2_n2.c:55
Вывод
Показать выходные данные от: Построение
1> slamch.c
1> slaswp.c
1> sscal.c
1> main.cpp
1> pSolver.cpp
1> xilink: executing 'link'
1> sSolver.vcxproj -> D:\sSolver\x64\Release\sSolver.exe
========== Перестроение всех: успешно: 1, с ошибками: 0, пропущено: 0 ==========
Готово
ENG
5:13
9/24/2013
Обозреватель серверов
Панель элементов


sSolver - Microsoft Visual Studio
Data race
Intel Inspector XE 2013
Target
Analysis Type
Collection Log
Summary
Sources
Write - Thread OMP Worker Thread #2 (4900) (sSolver.exe!sgetc2_ompparallel@45 - sgetc2_n2.c:58)
Write - Thread OMP Worker Thread #1 (3260) (sSolver.exe!sgetc2_ompparallel@45 - sgetc2_n2.c:58)
Call Stack
Вывод
Показать выходные данные от: Построение
1> slamch.c
1> slaswp.c
1> sscal.c
1> main.cpp
1> pSolver.cpp
1> xilink: executing 'link'
1> sSolver.vcxproj -> D:\sSolver\x64\Release\sSolver.exe
========== Перестроение всех: успешно: 1, с ошибками: 0, пропущено: 0 ==========
Готово

Intel® Advisor XE

Многоплатформенный: MS Windows, Linux.

Поддержка: C/C++, Fortran, .NET, C#.

Инструмент проектирования, помогающий распараллелить приложение.

Инструменты распараллеливания: OpenMP, Intel® TBB, Intel® Cilk™ Plus, Microsoft TPL (Task Parallel Library).

Моделирование поведения параллельного приложения до внедрения параллелизма.

Этапы распараллеливания

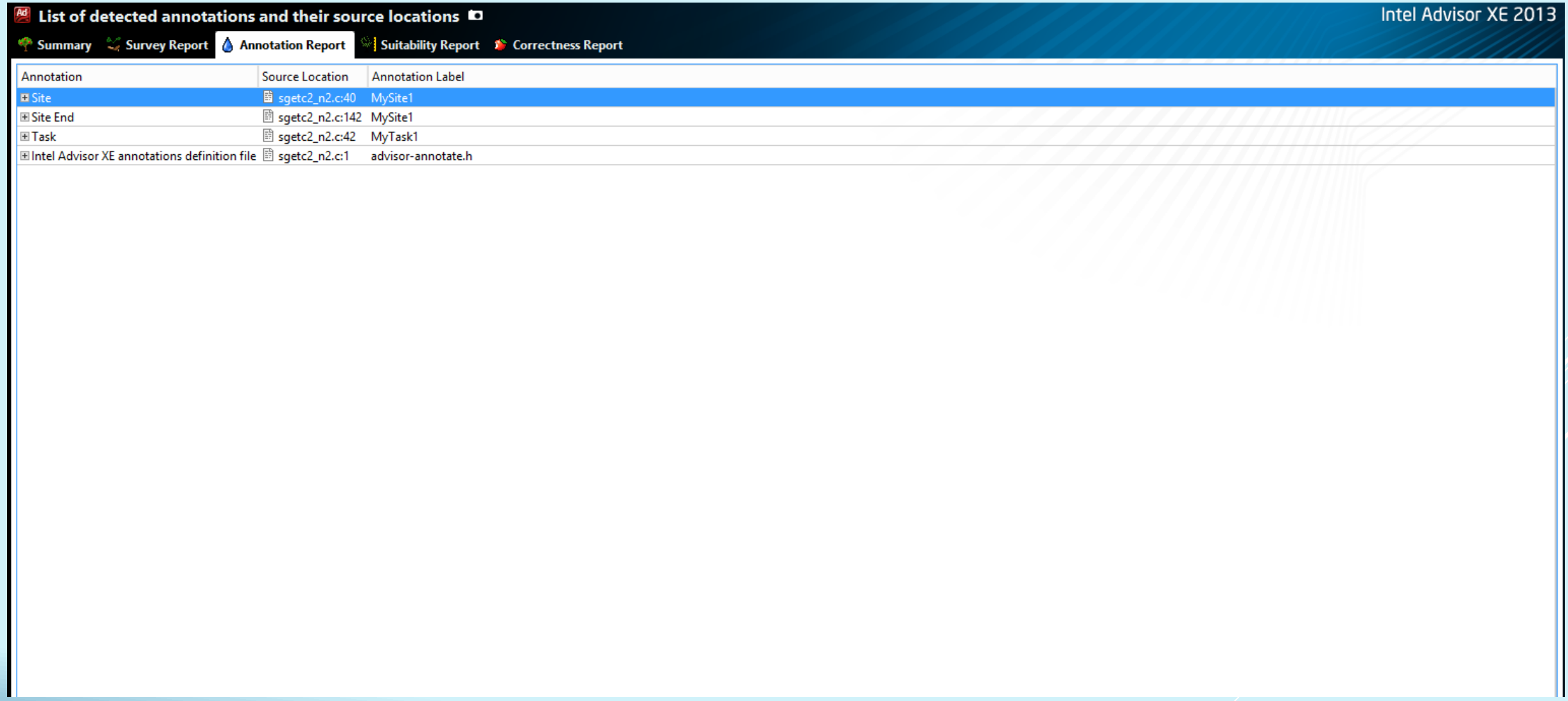
1. **Survey** – поиск фрагментов кода-кандидатов на распараллеливание.

2. **Annotations** – проектирование и описание структуры распараллеливания программы.

3. **Suitability** – моделирование поведения параллельной программы, оценка возможного выигрыша в эффективности в результате распараллеливания и масштабируемости.

4. **Correctness** – выявление возможных проблем при распараллеливании программы.

Реализация параллельного алгоритма «вручную».


List of detected annotations and their source locations
Intel Advisor XE 2013
Summary
Survey Report
Annotation Report
Suitability Report
Correctness Report
Annotation
Source Location
Annotation Label
Site
sgetc2_n2.c:40
MySite1
Site End
sgetc2_n2.c:142
MySite1
Task
sgetc2_n2.c:42
MyTask1
Intel Advisor XE annotations definition file
sgetc2_n2.c:1
advisor-annotate.h

Аннотации – макросы, с помощью которых описывается схема распараллеливания программы: параллельные секции, точки синхронизации и т.д., и которые использует Advisor при моделировании поведения параллельной программы. В дальнейшем аннотации могут быть заменены соответствующими конструкциями распараллеливания.

Аннотации не искажают поведения программы.

Виды аннотаций:

- аннотации, маркирующие начало и конец параллельных секций;
- аннотации, маркирующие задачи, которые будут исполняться параллельно;
- аннотации, маркирующие точки синхронизации;
- аннотации, включающие и отключающие сбор статистики (позволяют исключить из анализа неинтересные фрагменты кода);
- другие аннотации (используются реже, при проверке корректности).

Suitability анализ – использует аннотации, описывающие структуру распараллеливания программы для моделирования её поведения и прогноза ряда метрик.

Correctness анализ – использует аннотации, описывающие структуру распараллеливания программы для выявления возможных гонок за данными и других проблем.

Для того, чтобы использовать аннотации, необходимо:

1. добавить аннотации в исходный текст программы;
2. добавить заголовок `#include "advisor-annotate.h";`
3. добавить в параметрах компиляции путь к соответствующему include-каталогу.

Список аннотаций

`ANNOTATE_SITE_BEGIN(_SITE)`	– начало области анализа.
`ANNOTATE_SITE_END(...)`	– конец области анализа.
`ANNOTATE_TASK_BEGIN(_TASK)`	– начало области, ассоциированной с задачей.
`ANNOTATE_TASK_END(...)`	– конец области, ассоциированной с задачей.
`ANNOTATE_ITERATION_TASK(_TASK)`	– задачей является итерация цикла.
`ANNOTATE_LOCK_ACQUIRE(_ADDR)`	– захват блокировки.
`ANNOTATE_LOCK_RELEASE(_ADDR)`	– освобождение блокировки.
`ANNOTATE_RECORD_ALLOCATION(_ADDR, _SIZE)`	– запись размещения памяти пользователем.
`ANNOTATE_RECORD_DEALLOCATION(_ADDR)`	– запись освобождения памяти пользователем.
`ANNOTATE_INDUCTION_USES(_ADDR, _SIZE)`	– объявить область памяти (переменную) индуктивной.
`ANNOTATE_REDUCTION_USES(_ADDR, _SIZE)`	– объявить область памяти (переменную)редуктором.

Список аннотаций

`ANNOTATE_OBSERVE_USES(_ADDR, _SIZE)` – фиксировать все операции доступа к указанной области памяти.

`ANNOTATE_CLEAR_USES(_ADDR)` – завершить фиксацию всех операций доступа к указанной области памяти.

```
ANNOTATE_DISABLE_OBSERVATION_PUSH
ANNOTATE_DISABLE_OBSERVATION_POP
ANNOTATE_DISABLE_COLLECTION_PUSH
ANNOTATE_DISABLE_COLLECTION_POP
ANNOTATE_AGGREGATE_TASK(_COUNT)
```

Полезные советы

✓Аннотации следует размещать в отдельной строке.

✓Аннотации не следует размещать в макросах.

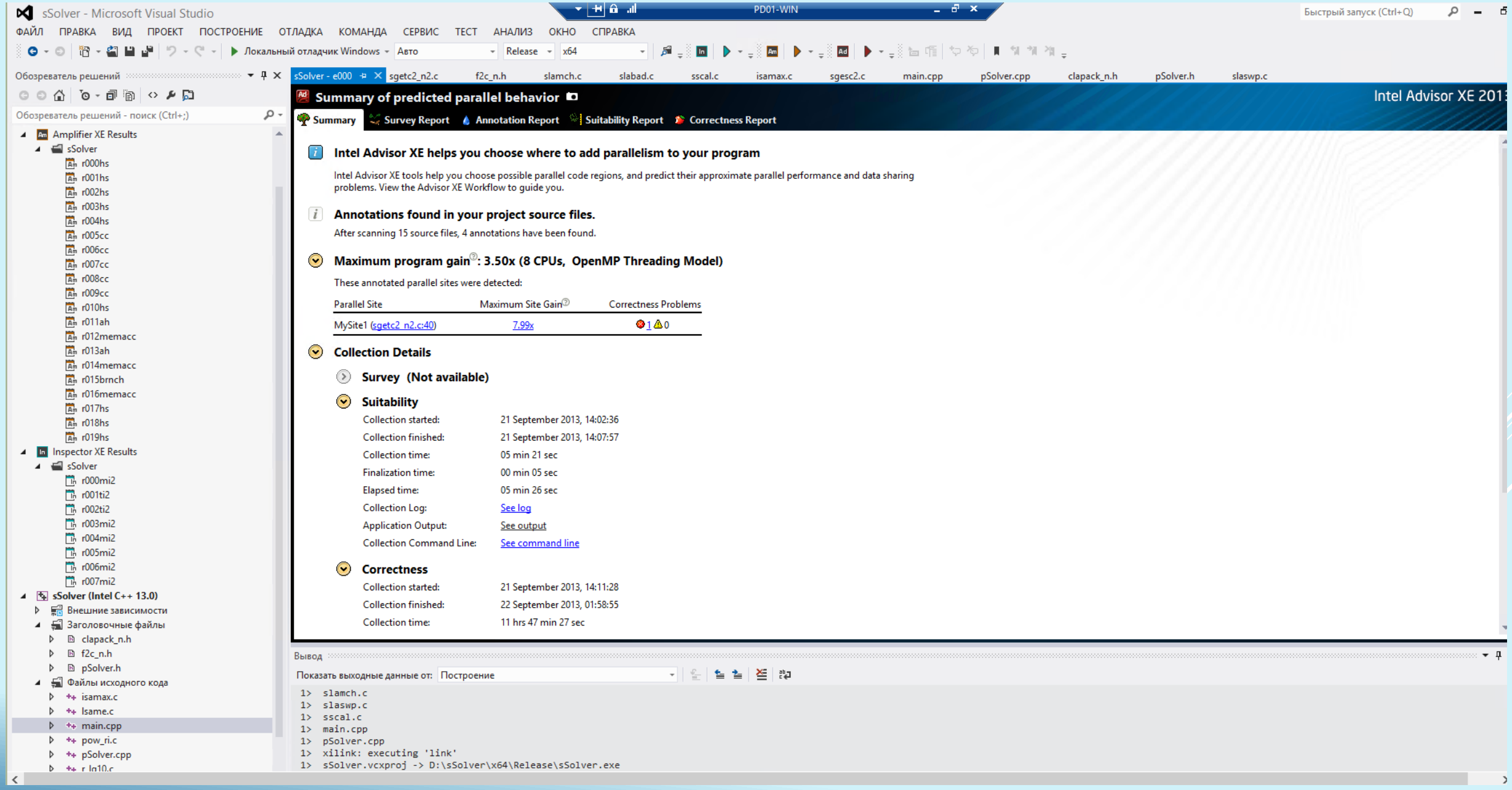
✓Для некоторых аннотаций важным является наличие в исходном коде парной аннотации.

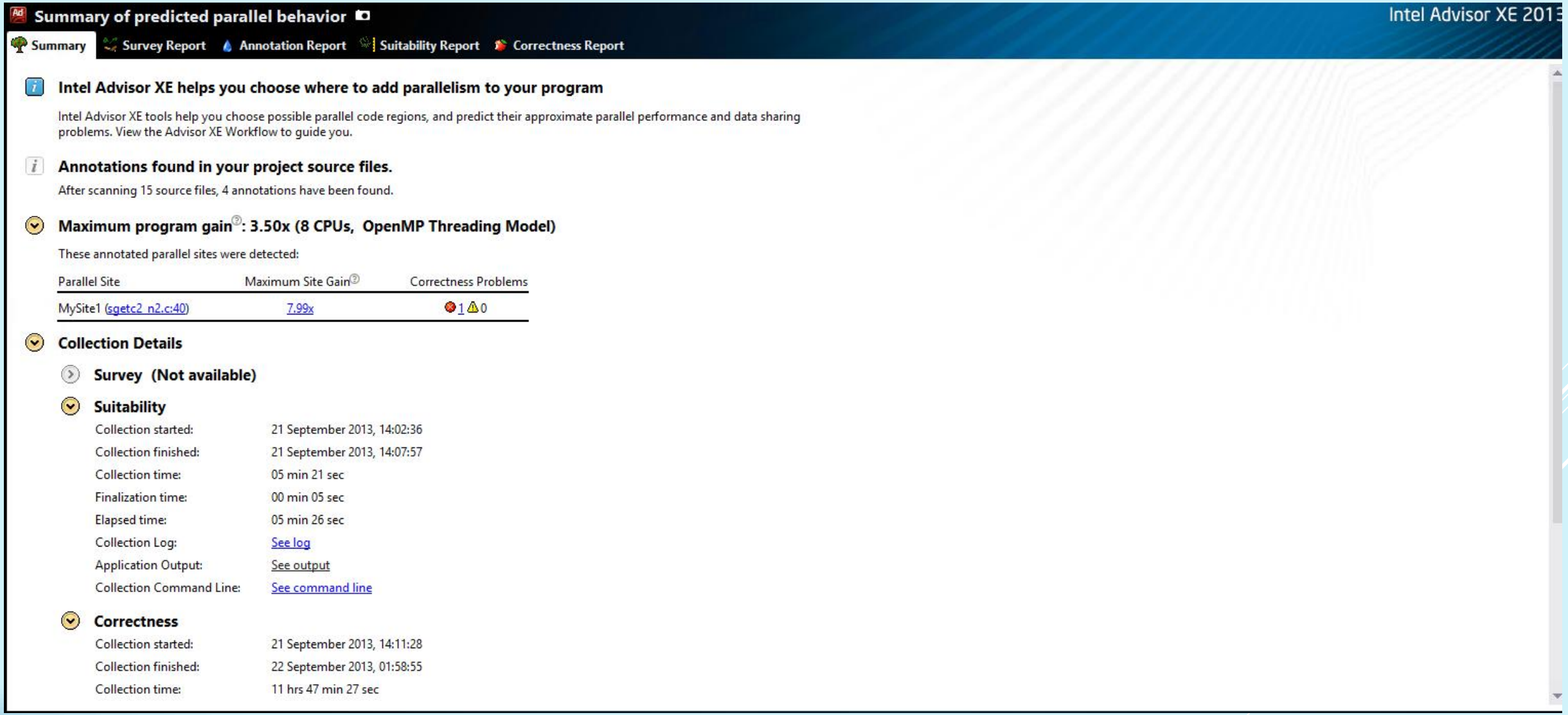
✓Одной стартовой аннотации могут соответствовать несколько финальных аннотаций.

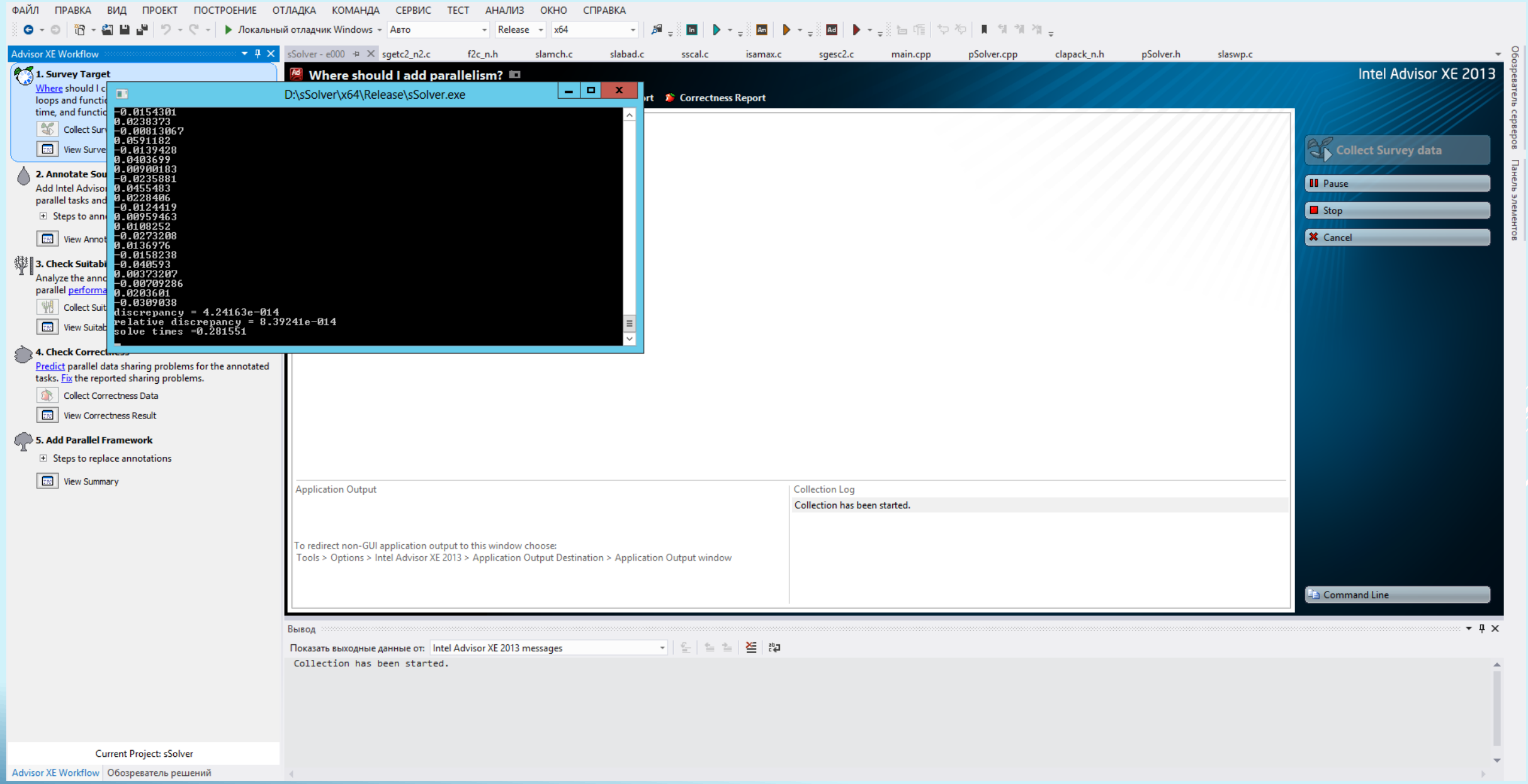
✓В ситуации совместной работы над проектом, когда разработчику **A** нужны аннотации, а разработчику **B** не нужны аннотации, можно использовать:

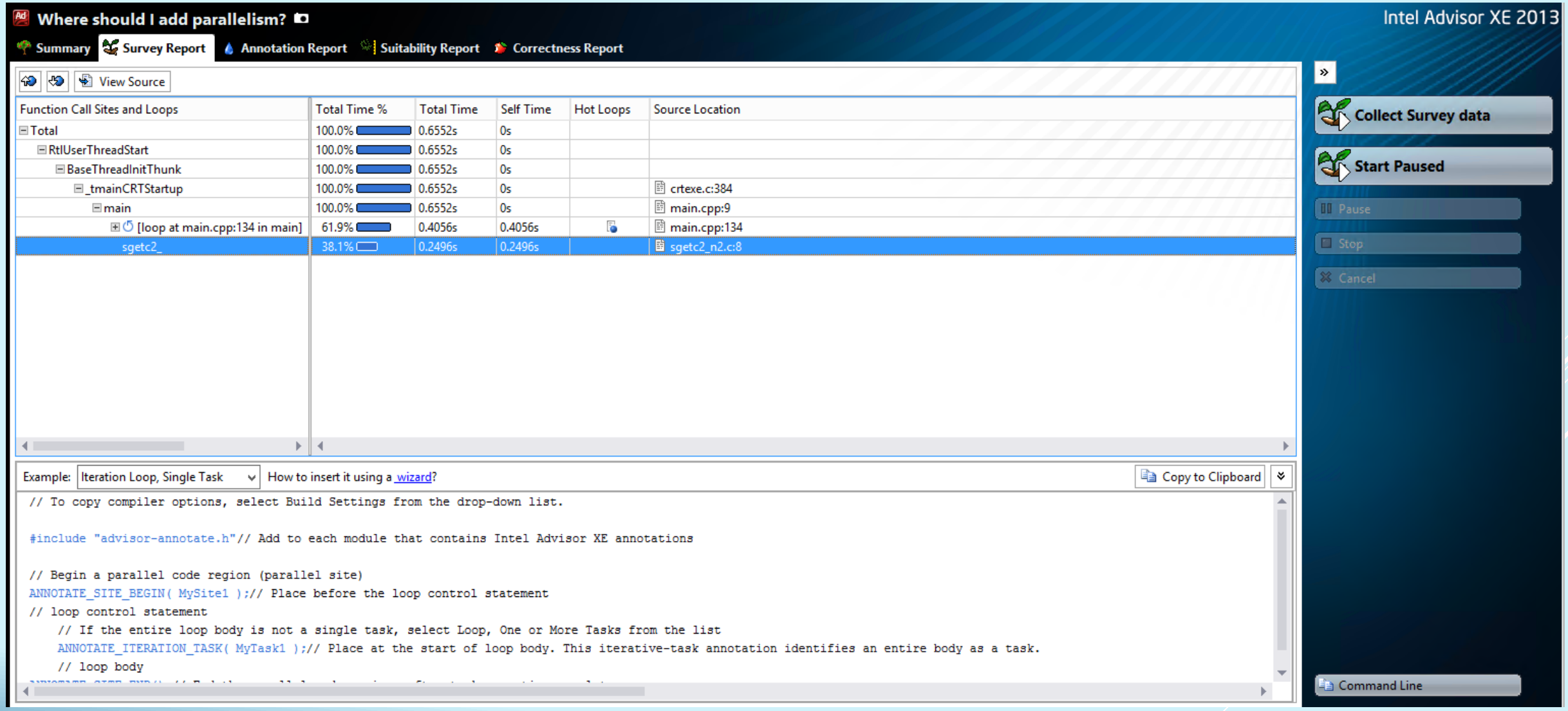
```
#define ANNOTATE_EXPAND_NULL

#include "advisor-annotate.h"
```

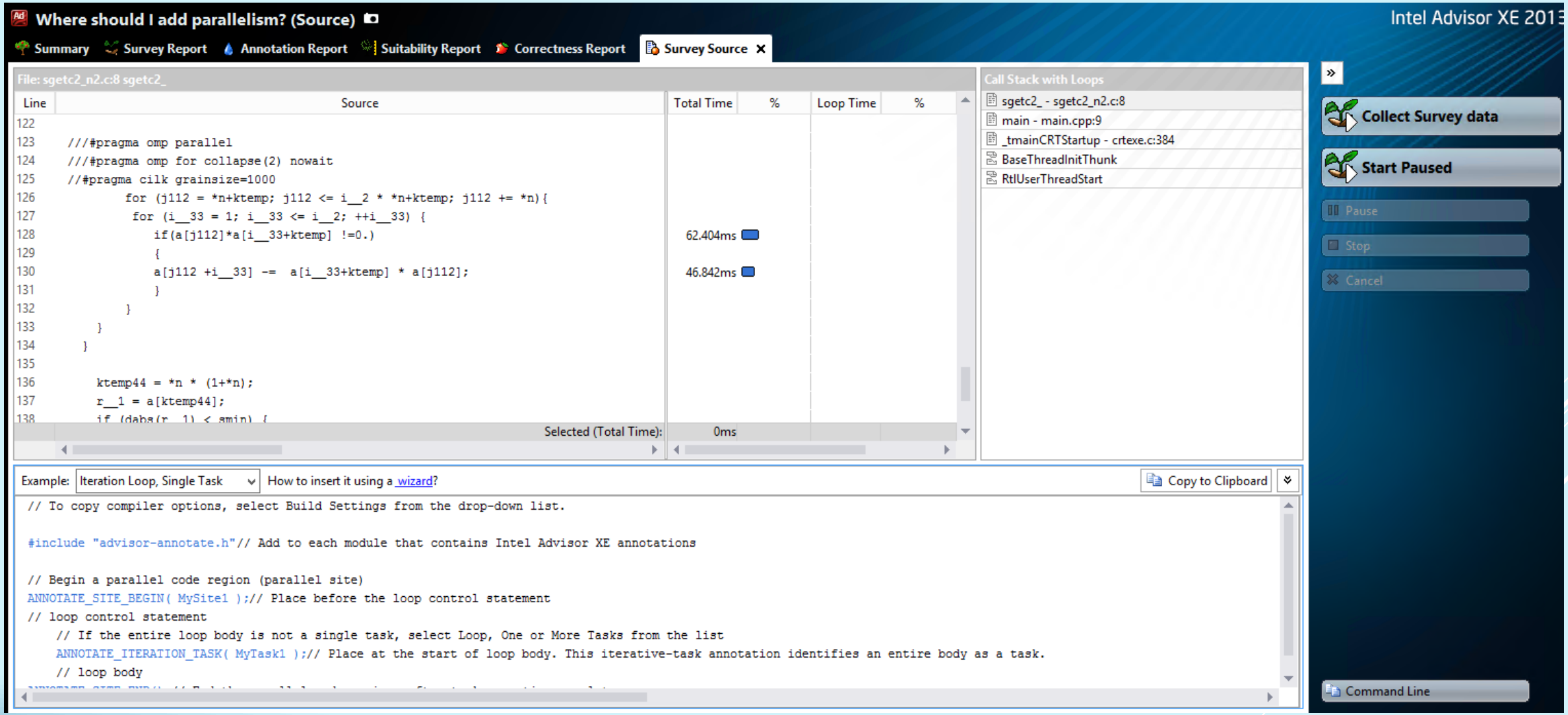
```
#include "advisor-annotate.h"
#include "f2c_n.h"
…
/* Function Body */
…
i__1 = *n - 1;
ktemp = 0;

ANNOTATE_SITE_BEGIN(MySite1);
for (i__ = 1; i__ <= i__1; ++i__) {//333
ANNOTATE_ITERATION_TASK(MyTask1);
ktemp += (1 + *n);
xmax = 0.e0f;
for (jp = i__; jp <= *n; ++jp) {//0
for (ip = i__; ip <= *n; ++ip) {//1
r__1 = dabs(a[ip + jp * *n]);
if (r__1 >= xmax) {//2
// Data race
xmax = r__1;
// Data race
…
ktemp44 = *n * (1+*n);
r__1 = a[ktemp44];
    if (dabs(r__1) < smin) {
*info = *n;
a[ktemp44] = smin;
    }
ANNOTATE_SITE_END(MySite1);

return 0;
} /* sgetc2_ */
```


sSolver - Microsoft Visual Studio
PD01-WIN
Быстрый запуск (Ctrl+Q)
ФАЙЛ ПРАВКА ВИД ПРОЕКТ ПОСТРОЕНИЕ ОТЛАДКА КОМАНДА СЕРВИС ТЕСТ АНАЛИЗ ОКНО СПРАВКА
Локальный отладчик Windows
Авто
Release
x64
Обозреватель решений
Обозреватель решений - поиск (Ctrl+;)
Amplifier XE Results
sSolver
r000hs
r001hs
r002hs
r003hs
r004hs
r005cc
r006cc
r007cc
r008cc
r009cc
r010hs
r011ah
r012memacc
r013ah
r014memacc
r015brnch
r016memacc
r017hs
r018hs
r019hs
Inspector XE Results
sSolver
r000mi2
r001ti2
r002ti2
r003mi2
r004mi2
r005mi2
r006mi2
r007mi2
sSolver (Intel C++ 13.0)
Внешние зависимости
Заголовочные файлы
clapack_n.h
f2c_n.h
pSolver.h
Файлы исходного кода
isamax.c
lsame.c
main.cpp
pow_ri.c
pSolver.cpp
sSolver - e000
sgetc2_n2.c
f2c_n.h
slamch.c
slabad.c
sscal.c
isamax.c
sgesc2.c
main.cpp
pSolver.cpp
clapack_n.h
pSolver.h
slaswp.c
Summary of predicted parallel behavior
Intel Advisor XE 201
Summary
Survey Report
Annotation Report
Suitability Report
Correctness Report
Intel Advisor XE helps you choose where to add parallelism to your program
Intel Advisor XE tools help you choose possible parallel code regions, and predict their approximate parallel performance and data sharing problems. View the Advisor XE Workflow to guide you.
Annotations found in your project source files.
After scanning 15 source files, 4 annotations have been found.
Maximum program gain: 3.50x (8 CPUs, OpenMP Threading Model)
These annotated parallel sites were detected:
Parallel Site
Maximum Site Gain
Correctness Problems
MySite1 (sgetc2_n2.c:40)
7.99x
1 0
Collection Details
Survey (Not available)
Suitability
Collection started: 21 September 2013, 14:02:36
Collection finished: 21 September 2013, 14:07:57
Collection time: 05 min 21 sec
Finalization time: 00 min 05 sec
Elapsed time: 05 min 26 sec
Collection Log: See log
Application Output: See output
Collection Command Line: See command line
Correctness
Collection started: 21 September 2013, 14:11:28
Collection finished: 22 September 2013, 01:58:55
Collection time: 11 hrs 47 min 27 sec
Вывод
Показать выходные данные от: Построение
1> slamch.c
1> slaswp.c
1> sscal.c
1> main.cpp
1> pSolver.cpp
1> xilink: executing 'link'
1> sSolver.vcxproj -> D:\sSolver\x64\Release\sSolver.exe


Summary of predicted parallel behavior
Intel Advisor XE 201
Summary
Survey Report
Annotation Report
Suitability Report
Correctness Report
Intel Advisor XE helps you choose where to add parallelism to your program
Intel Advisor XE tools help you choose possible parallel code regions, and predict their approximate parallel performance and data sharing problems. View the Advisor XE Workflow to guide you.
Annotations found in your project source files.
After scanning 15 source files, 4 annotations have been found.
Maximum program gain: 3.50x (8 CPUs, OpenMP Threading Model)
These annotated parallel sites were detected:
Parallel Site
Maximum Site Gain
Correctness Problems
MySite1 (sgetc2_n2.c:40)
7.99x
1 0
Collection Details
Survey (Not available)
Suitability
Collection started: 21 September 2013, 14:02:36
Collection finished: 21 September 2013, 14:07:57
Collection time: 05 min 21 sec
Finalization time: 00 min 05 sec
Elapsed time: 05 min 26 sec
Collection Log: See log
Application Output: See output
Collection Command Line: See command line
Correctness
Collection started: 21 September 2013, 14:11:28
Collection finished: 22 September 2013, 01:58:55
Collection time: 11 hrs 47 min 27 sec


ФАЙЛ ПРАВКА ВИД ПРОЕКТ ПОСТРОЕНИЕ ОТЛАДКА КОМАНДА СЕРВИС ТЕСТ АНАЛИЗ ОКНО СПРАВКА
Локальный отладчик Windows
Авто
Release
x64
Advisor XE Workflow
1. Survey Target
Collect Survey data
View Survey Report
2. Annotate Sources
Add Intel Advisor annotations to identify parallel tasks and data sharing
Steps to annotate
View Annotation Report
3. Check Suitability
Analyze the annotated parallel performance
Collect Suitability Data
View Suitability Report
4. Check Correctness
Predict parallel data sharing problems for the annotated tasks. Fix the reported sharing problems.
Collect Correctness Data
View Correctness Result
5. Add Parallel Framework
Steps to replace annotations
View Summary
Current Project: sSolver
Advisor XE Workflow
Обозреватель решений
sSolver - e000
sgetc2_n2.c
f2c_n.h
slamch.c
slabad.c
sscal.c
isamax.c
sgesc2.c
main.cpp
pSolver.cpp
clapack_n.h
pSolver.h
slaswp.c
Where should I add parallelism?
Correctness Report
Intel Advisor XE 2013
Collect Survey data
Pause
Stop
Cancel
Command Line
Application Output
To redirect non-GUI application output to this window choose:
Tools > Options > Intel Advisor XE 2013 > Application Output Destination > Application Output window
Collection Log
Collection has been started.
D:\sSolver\x64\Release\sSolver.exe
-0.0154301
0.0238373
-0.00813067
0.0591182
-0.0139428
0.0403699
0.00900183
-0.0235881
0.0455483
0.0228406
-0.0124419
0.00959463
0.0108252
-0.0273208
0.0136976
-0.0158238
-0.040593
0.00373207
-0.00709286
0.0203601
-0.0309038
discrepancy = 4.24163e-014
relative discrepancy = 8.39241e-014
solve times =0.281551
Вывод
Показать выходные данные от: Intel Advisor XE 2013 messages
Collection has been started.
Обозреватель серверов
Панель элементов


Where should I add parallelism?
Intel Advisor XE 2013
Summary
Survey Report
Annotation Report
Suitability Report
Correctness Report
View Source
Function Call Sites and Loops
Total Time %
Total Time
Self Time
Hot Loops
Source Location
Total 100.0% 0.6552s 0s
RtlUserThreadStart 100.0% 0.6552s 0s
BaseThreadInitThunk 100.0% 0.6552s 0s
_tmainCRTStartup 100.0% 0.6552s 0s crtexe.c:384
main 100.0% 0.6552s 0s main.cpp:9
[loop at main.cpp:134 in main] 61.9% 0.4056s 0.4056s main.cpp:134
sgetc2_ 38.1% 0.2496s 0.2496s sgetc2_n2.c:8
Collect Survey data
Start Paused
Pause
Stop
Cancel
Command Line
Example: Iteration Loop, Single Task
How to insert it using a wizard?
Copy to Clipboard
// To copy compiler options, select Build Settings from the drop-down list.
#include "advisor-annotate.h"// Add to each module that contains Intel Advisor XE annotations
// Begin a parallel code region (parallel site)
ANNOTATE_SITE_BEGIN(MySite1);// Place before the loop control statement
// loop control statement
// If the entire loop body is not a single task, select Loop, One or More Tasks from the list
ANNOTATE_ITERATION_TASK(MyTask1);// Place at the start of loop body. This iterative-task annotation identifies an entire body as a task.
// loop body


Where should I add parallelism? (Source)
Intel Advisor XE 2013
Summary
Survey Report
Annotation Report
Suitability Report
Correctness Report
Survey Source
File: sgetc2_n2.c:8 sgetc2_
Line
Source
Total Time
%
Loop Time
%
122
123 ///#pragma omp parallel
124 ///#pragma omp for collapse(2) nowait
125 //#pragma cilk grainsize=1000
126 for (j112 = *n+ktemp; j112 <= i__2 * *n+ktemp; j112 += *n){
127 for (i__33 = 1; i__33 <= i__2; ++i__33) {
128 if(a[j112]*a[i__33+ktemp] !=0.)
129 {
130 a[j112 +i__33] -= a[i__33+ktemp] * a[j112];
131 }
132 }
133 }
134 }
135
136 ktemp44 = *n * (1+*n);
137 r__1 = a[ktemp44];
138 if (daba(r__1) < smin) {
62.404ms
46.842ms
Selected (Total Time):
0ms
Call Stack with Loops
sgetc2_ - sgetc2_n2.c:8
main - main.cpp:9
_tmainCRTStartup - crtexe.c:384
BaseThreadInitThunk
RtlUserThreadStart
Collect Survey data
Start Paused
Pause
Stop
Cancel
Command Line
Example:
Iteration Loop, Single Task
How to insert it using a wizard?
Copy to Clipboard
// To copy compiler options, select Build Settings from the drop-down list.
#include "advisor-annotate.h"// Add to each module that contains Intel Advisor XE annotations
// Begin a parallel code region (parallel site)
ANNOTATE_SITE_BEGIN(MySite1);// Place before the loop control statement
// loop control statement
// If the entire loop body is not a single task, select Loop, One or More Tasks from the list
ANNOTATE_ITERATION_TASK(MyTask1);// Place at the start of loop body. This iterative-task annotation identifies an entire body as a task.
// loop body

What are the performance implications of the annotated sites?

Intel Advisor XE 201

Summary | Survey Report | Annotation Report | Suitability Report | Correctness Report

All Sites

Maximum Program Gain For All Sites:

3.50x

Target CPU Count: 8 Threading Model: OpenMP

Annotation Label	Source Location	Maximum Site Gain	Maximum Total Gain	Average Instance Time	Total Time
MySite1	sgetc2_n2.c:40	7.99x	3.50x	260.4731s	260.4731s

Collect Suitability data

Start Paused

Pause

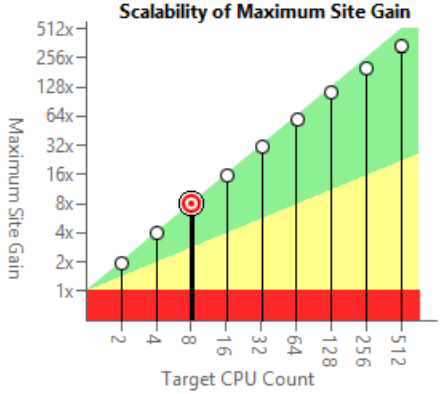
Stop

Cancel

Command Line

Selected Site

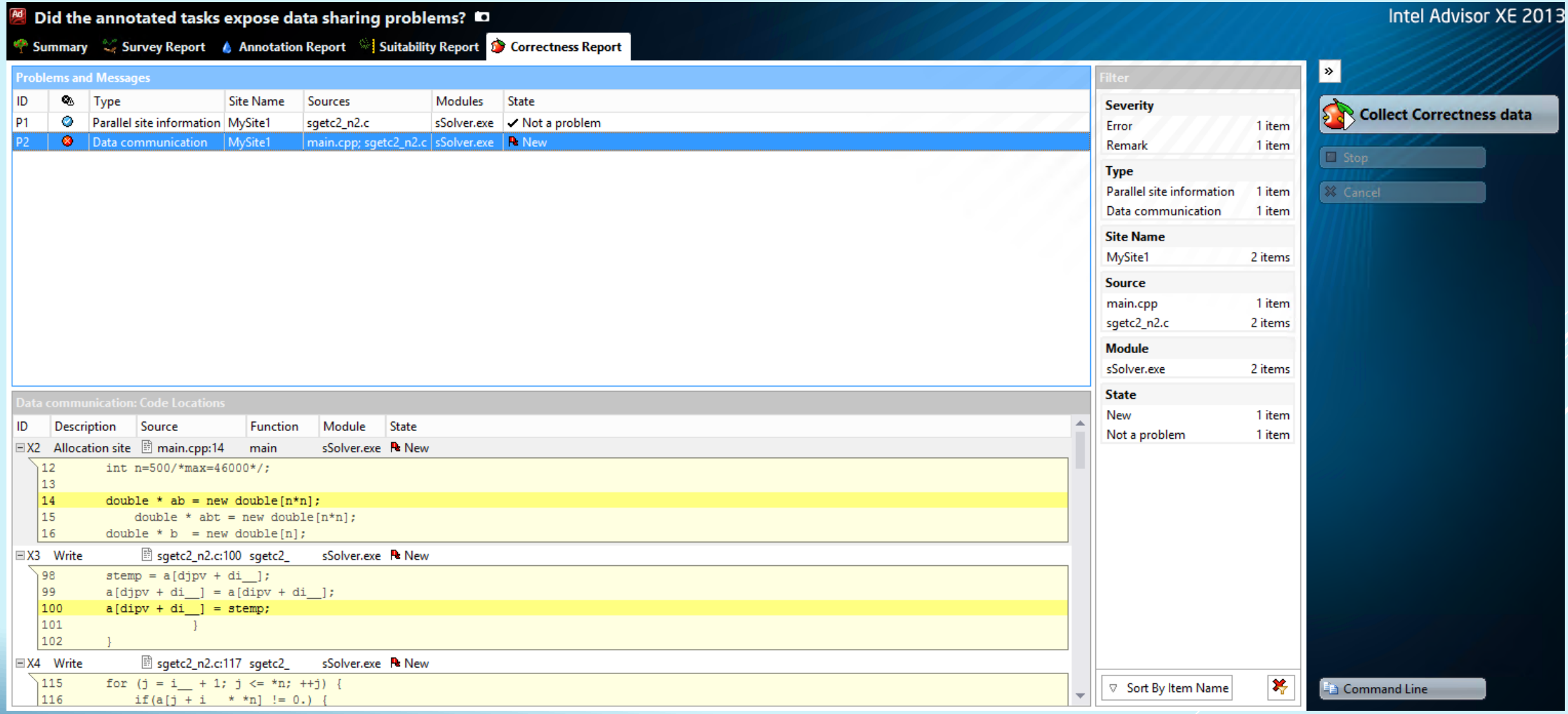
Scalability of Maximum Site Gain

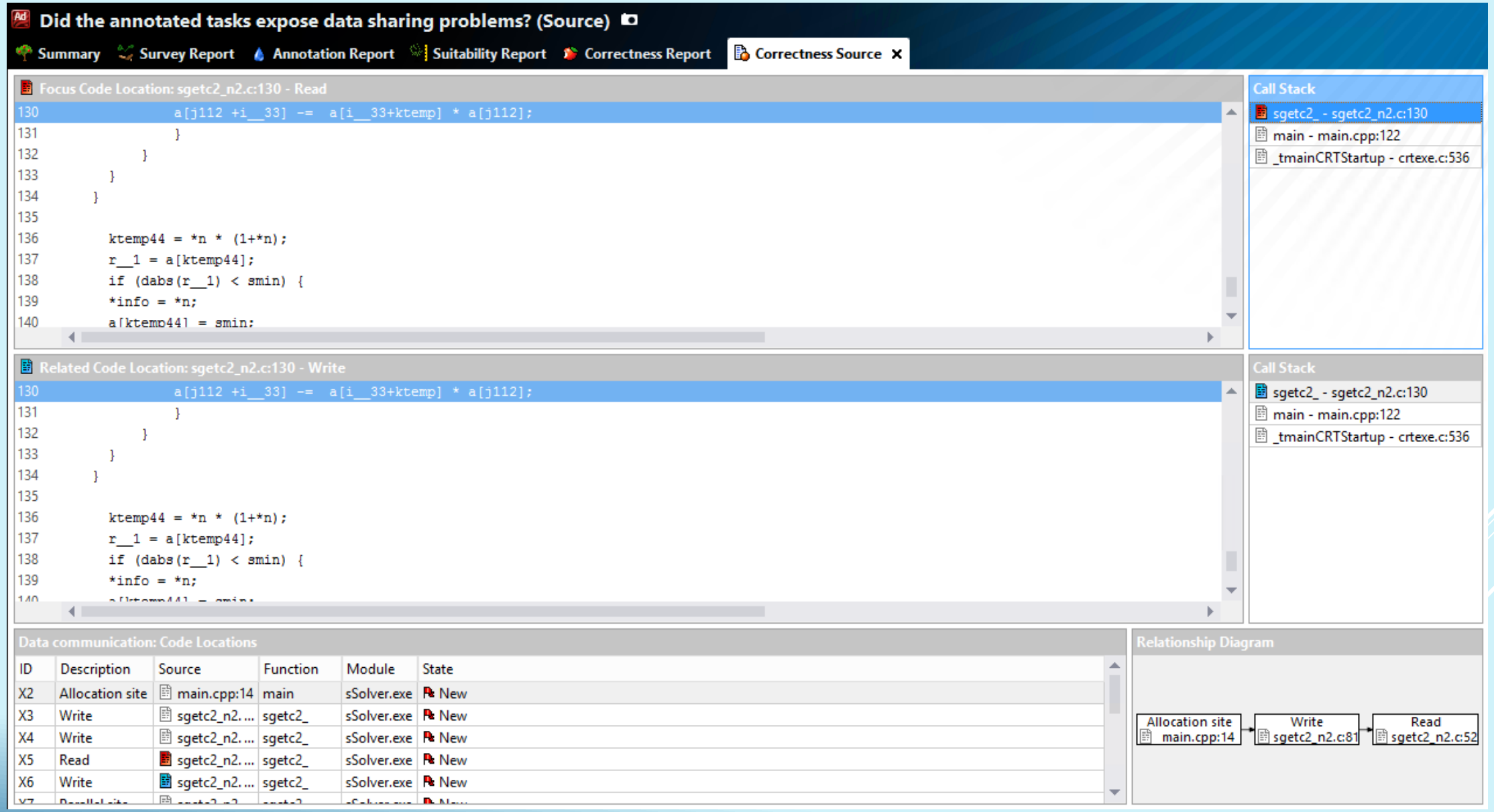


Changes I will make to this site to improve performance

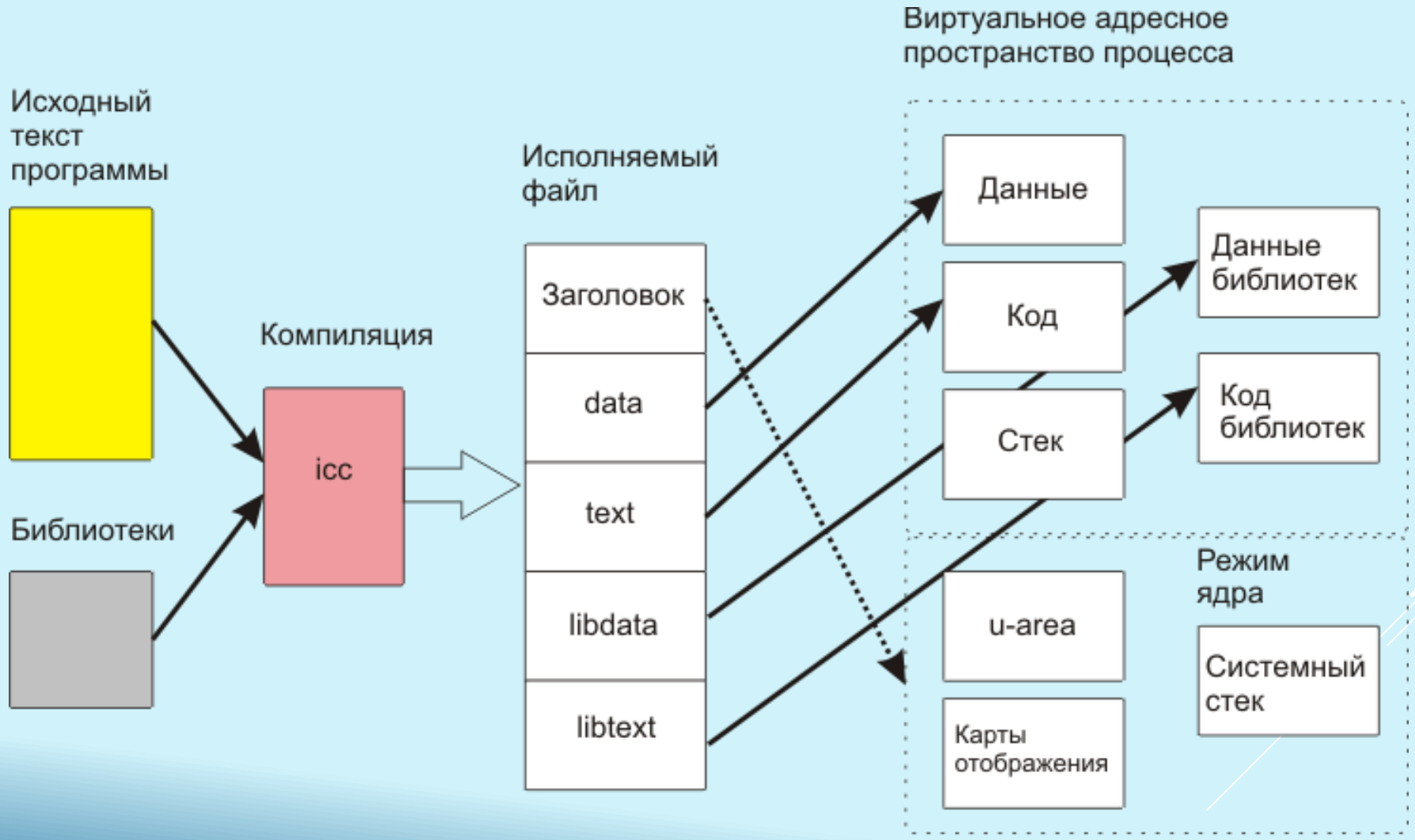
Type of Change	Benefit if Checked	Loss if Unchecked	Recommended
☑ Reduce Site Overhead			No
☑ Reduce Task Overhead			No
☑ Reduce Lock Overhead			No
☑ Reduce Lock Contention			No
☑ Enable Task Chunking			No

Annotation	Annotation Label	Source Location	Number of Instances	Maximum Instance Time	Average Instance Time	Minimum Instance Time	Total Time
Selected Site	MySite1	sgetc2_n2.c:40	1	260.4731s	260.4731s	260.4731s	260.4731s
Task	MyTask1	sgetc2_n2.c:42	4,999	0.3044s	0.0521s	< 0.0001s	260.4693s


Did the annotated tasks expose data sharing problems?
Intel Advisor XE 2013
Summary
Survey Report
Annotation Report
Suitability Report
Correctness Report
Problems and Messages
ID
Type
Site Name
Sources
Modules
State
P1
Parallel site information
MySite1
sgetc2_n2.c
sSolver.exe
Not a problem
P2
Data communication
MySite1
main.cpp; sgetc2_n2.c
sSolver.exe
New
Filter
Severity
Error 1 item
Remark 1 item
Type
Parallel site information 1 item
Data communication 1 item
Site Name
MySite1 2 items
Source
main.cpp 1 item
sgetc2_n2.c 2 items
Module
sSolver.exe 2 items
State
New 1 item
Not a problem 1 item
Collect Correctness data
Stop
Cancel
Data communication: Code Locations
ID
Description
Source
Function
Module
State
X2 Allocation site main.cpp:14 main sSolver.exe New
12 int n=500/*max=46000*/;
13
14 double * ab = new double[n*n];
15 double * abt = new double[n*n];
16 double * b = new double[n];
X3 Write sgetc2_n2.c:100 sgetc2_ sSolver.exe New
98 stemp = a[djpv + di__];
99 a[djpv + di__] = a[dipv + di__];
100 a[dipv + di__] = stemp;
101 }
102 }
X4 Write sgetc2_n2.c:117 sgetc2_ sSolver.exe New
115 for (j = i__ + 1; j <= *n; ++j) {
116 if(a[j + i * *n] != 0.) {
Sort By Item Name
Command Line


Did the annotated tasks expose data sharing problems? (Source)
Summary
Survey Report
Annotation Report
Suitability Report
Correctness Report
Correctness Source
Focus Code Location: sgetc2_n2.c:130 - Read
130 a[j112 +i__33] -= a[i__33+ktemp] * a[j112];
131 }
132 }
133 }
134 }
135
136 ktemp44 = *n * (1+*n);
137 r__1 = a[ktemp44];
138 if (dabs(r__1) < smin) {
139 *info = *n;
140 a[ktemp44] = smin;
Call Stack
sgetc2_ - sgetc2_n2.c:130
main - main.cpp:122
_tmainCRTStartup - crtexe.c:536
Related Code Location: sgetc2_n2.c:130 - Write
130 a[j112 +i__33] -= a[i__33+ktemp] * a[j112];
131 }
132 }
133 }
134 }
135
136 ktemp44 = *n * (1+*n);
137 r__1 = a[ktemp44];
138 if (dabs(r__1) < smin) {
139 *info = *n;
Call Stack
sgetc2_ - sgetc2_n2.c:130
main - main.cpp:122
_tmainCRTStartup - crtexe.c:536
Data communication: Code Locations
ID Description Source Function Module State
X2 Allocation site main.cpp:14 main sSolver.exe New
X3 Write sgetc2_n2. ... sgetc2_ sSolver.exe New
X4 Write sgetc2_n2. ... sgetc2_ sSolver.exe New
X5 Read sgetc2_n2. ... sgetc2_ sSolver.exe New
X6 Write sgetc2_n2. ... sgetc2_ sSolver.exe New
Relationship Diagram
Allocation site main.cpp:14
Write sgetc2_n2.c:81
Read sgetc2_n2.c:52

Компиляторная оптимизация


Виртуальное адресное пространство процесса
Исходный текст программы
Исполняемый файл
Данные
Заголовок
Код
Данные библиотек
Компиляция
data
Стек
Код библиотек
icc
text
Библиотеки
Режим ядра
u-area
libdata
Системный стек
libtext
Карты отображения

GNU Compiler Collection (GCC) - компиляторы C/C++, Объектный C, Fortran, Ada, Java. Поддерживается платформа Linux. Распространяется свободно, в том числе в исходных кодах.

Поддерживаются различные варианты оптимизации, отладки, кодогенерации, компиляция для разных платформ.

Поддерживается Intel® Cilk™ Plus.

Ссылка на ресурс:

http://gcc.gnu.org/

Oracle Solaris Studio - компиляторы C/C++ и Fortran. Поддерживаются платформы Linux и Solaris, архитектуры x86 и SPARC.

Поддерживаются различные варианты оптимизации, отладки, кодогенерации.

Ссылка на ресурс:

http://www.oracle.com

PGI Workstation - компиляторы C/C++ и Fortran. Поддерживаются платформы Linux, Microsoft Windows и MacOS, архитектуры x86, AMD и CUDA.

Поддерживаются различные варианты оптимизации, автоматического распараллеливания, отладки, интеграция с Microsoft Visual Studio.

Fortran 2003.

Ссылка на ресурс:

http://www.pgroup.com/

NAG (Numerical Algorithms Group) Fortran Compiler - компилятор Fortran. Поддерживаются платформы Unix, Microsoft Windows и MacOS.

Поддерживаются различные варианты оптимизации, автоматического распараллеливания, отладки.

Fortran 2003/2008.

Ссылка на ресурс:

http://www.nag.co.uk

Microsoft Visual C++ - поддерживается платформа Microsoft Windows.

Интеграция в Microsoft Visual Studio.

Ссылка на ресурс:

http://www.microsoft.com

MinGW (Minimal GNU for Windows)

Ссылка на ресурс:

http://www.mingw.org/

Open64

Ссылка на ресурс:

http://www.open64.net

Clang

Ссылка на ресурс:

http://clang.llvm.org

и другие.

Intel® Composer XE включает компиляторы C/C++, Fortran, а также библиотеки.

Поддерживаются платформы Microsoft Windows и Linux.

Поддерживается работа с оптимизированными библиотеками: Intel®MKL, Intel®IPP.

Средства поддержки оптимизации.

Улучшенная поддержка векторизации (увеличенная разрядность векторных инструкций).

Интеграция в среды разработки Microsoft Visual Studio, Eclipse, XCode.

Совместимость с Microsoft Visual C, компиляторами GCC (Linux) и MacOS.

Поддержка Fortran 77 – 2003. Поддержка COARRAY и DO CONCURRENT из Fortran 2008.

Подробная диагностика.

Поддержка Intel® Cilk™ Plus

Intel® Cilk™ Plus – средство разработки параллельных программ, включающее небольшой набор ключевых слов, гиперобъекты, средства работы с массивами (расширенная индексная нотация), эффективную поддержку векторизации и другое.

КАКУЮ ОПТИМИЗАЦИЮ МОЖЕТ ВЫПОЛНИТЬ КОМПИЛЯТОР

Удаление общих подвыражений

Развёртка цикла

Перестановка циклов

Удаление инвариантных выражений из тела цикла

Подстановка функций

Свёртка и распространение констант

Исключение указателей

Объединение ветвей

Алгебраическая и логическая редукция

Другие виды автоматической оптимизации

- Оптимизация под архитектуру.
- Распараллеливание.
- Оптимизация с профилированием.
- Межпроцедурная оптимизация.

и т.д.

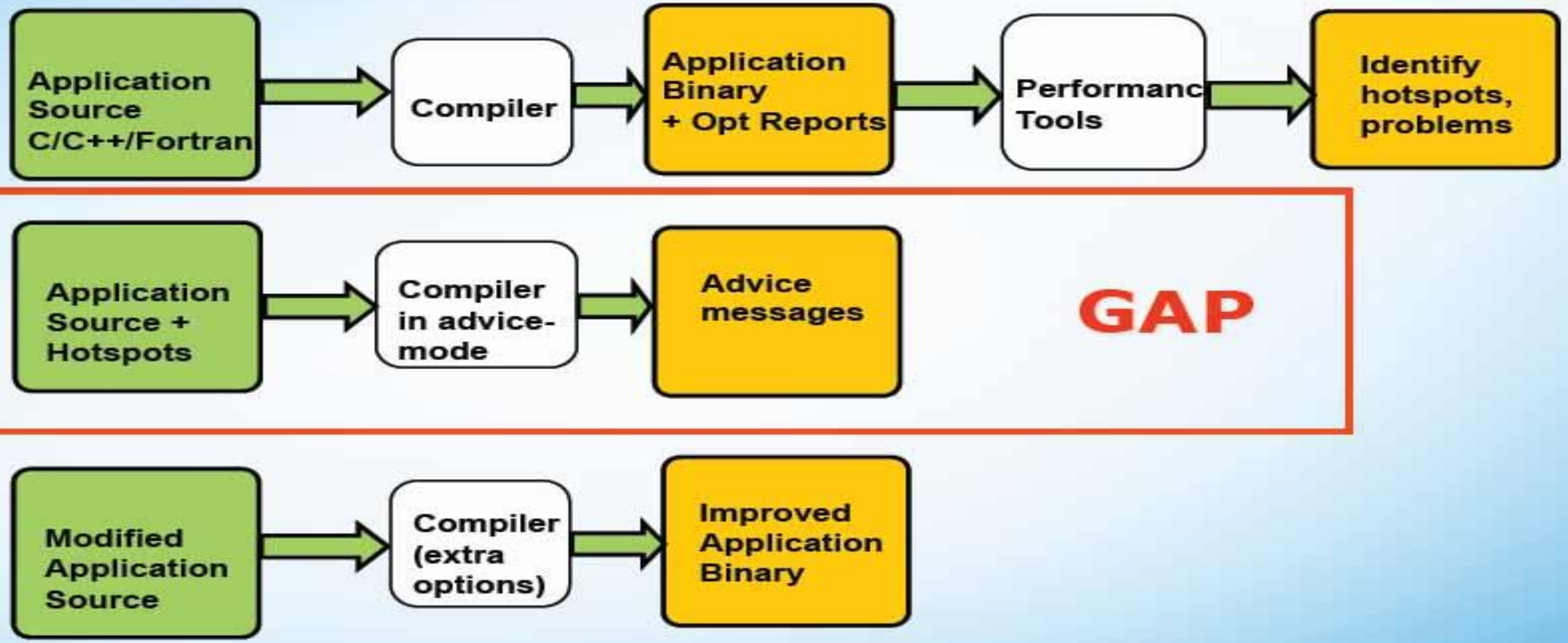
Отчёты об оптимизации

- /Qvec-report – генерация протокола векторизации (что векторизовано, что нет и почему).
- /Qpar-report - генерация протокола распараллеливания.
- /Qopt-report – генерация протокола оптимизации.

Профилирование на уровне циклов

Сбор статистики по циклам и функциям

- /Qprofile-loops:all - сбор статистики.
- LoopProfileViewer - утилита для просмотра отчётов.
- GAP - Guided Auto Parallelism (направляемая автопараллелизация).
- /Qguide (-guide) - запуск анализа.
- Не порождает параллельный код, но даёт рекомендации.


Application Source C/C++/Fortran
Compiler
Application Binary + Opt Reports
Performanc Tools
Identify hotspots, problems
Application Source + Hotspots
Compiler in advice-mode
Advice messages
GAP
Modified Application Source
Compiler (extra options)
Improved Application Binary

`/O1 (Windows), -O1 (Linux)`

- Глобальная оптимизация.
- Не увеличивает размер кода.

Для компилятора Microsoft уровень O1 эквивалентен набору оптимизаций:

- `/Og` - локальное и глобальное исключение общей части выражения, автоматическое выделение регистров (этот вид оптимизации позволяет компилятору хранить часто используемые переменные и части выражения в регистрах; ключевое слово register игнорируется), удаление инвариантов циклов, некоторые другие.
- `/Os` – оптимизация размера EXE-файлов, даже если это приведёт к уменьшению скорости выполнения программы.
- `/Oy` – подавляет создание указателей фрейма в стеке вызовов, что повышает скорость вызова функций, для хранения часто используемых значений и подвыражений используется выделенный для этого регистр (EBP для платформы x86). Этот вид оптимизации может затруднять отладку.
- `/Ob2` – управление подстановкой функций, 2-й уровень. Подстановка выполняется для функций, помеченных как `inline` или `__inline`, а также любых других функций, выбираемых компилятором, если он сочтёт это необходимым.
- `/Gs` – управление стековыми зондами. Стековый зонд является последовательностью кода, который компилятор вставляет в каждый вызов функции. Когда стековый зонд активирован, он занимает количество памяти, требуемое для хранения связанных локальных переменных функции.
- `/GF` – исключение повторяющихся строк. Разрешает компилятору создавать одиночные копии одинаковых строк в программном образе и в памяти при запуске.
- `/Gy` - позволяет компилятору упаковывать отдельные функции в форме упакованных функций (COMDAT).

`/O2 (Windows), -O2 (Linux)`

- Увеличивает размер кода.
- Оптимизация времени выполнения.
- Опция по умолчанию.
- Подстановки в коде.
- Развертывание циклов.
- Векторизация.

Для компилятора Microsoft уровень O2 эквивалентен набору оптимизаций:

- `/Og` – как в /O1.
- `/Oi` – заменяет вызов некоторых функций на встроенные или какие-либо другие формы функции, которые способствуют более быстрому выполнению приложения. Программы, использующие встроенные функции, выполняются быстрее, поскольку у них нет дополнительных издержек на вызов функции, но могут быть большего размера из-за создания дополнительного кода. Не все библиотечные функции могут использоваться как встроенные.
- `/Ot` – оптимизация не по размеру, а по скорости выполнения.
- `/Ob2` – как в /O1.
- `/Gs` – как в /O1.
- `/GF` – как в /O1.
- `/Gy` – как в /O1.

/O3 (Windows), -O3 (Linux)

- Высокоуровневая оптимизация.
- /O2 + более агрессивные методы.
- Улучшенная векторизация.
- Более полный учёт свойств циклов и массивов.
- Оптимизация циклов: разделение циклов (loop distribution), перестановка циклов (loop interchange), слияние циклов (loop fusion), развёртка циклов (loop unrolling).
- Подстановка кода в ветвлениях.
- Оптимизация под размер кэша.
- Предвыборка и предсказания ветвления.
- Возможна большая эффективность в приложениях, включающих обработку больших массивов.

Оптимизация под архитектуру

`/Qax (Windows), -ax (Linux)`

Оптимизация под архитектуру Intel

- QxHost
- QxAVX
- QxSSE2, QxSSE3, QxSSE3_ATOM, QxSSE4.1, QxSSE4.2, QxSSSE3

Автоматическое распараллеливание

`/Qparallel (Windows), -parallel (Linux)`

- Автоматическое распараллеливание.
- Определяются те части кода, которые можно распараллелить.
- Выполняется анализ зависимостей.
- Разделение данных для параллельной обработки.
- Работа с циклами.

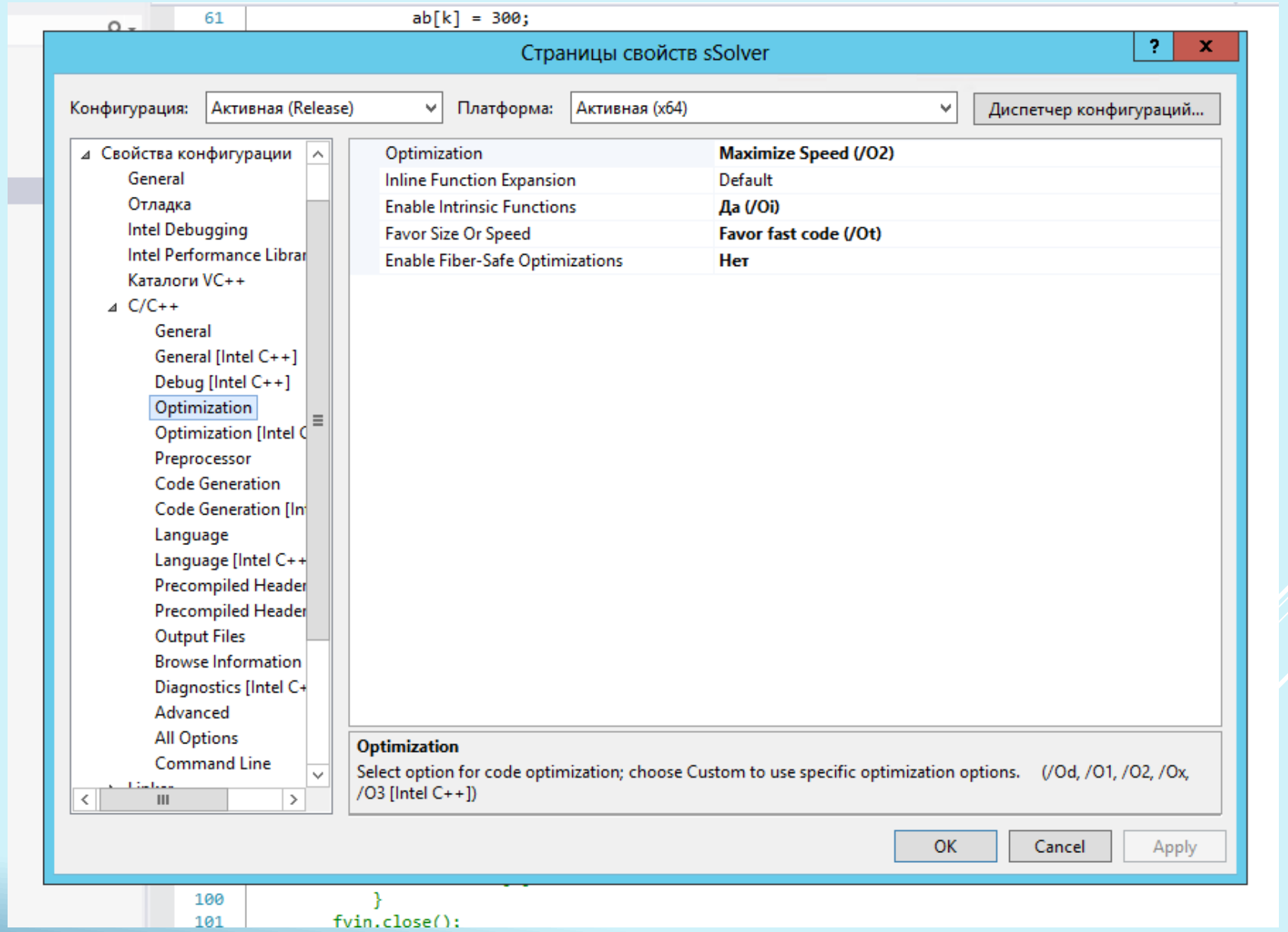
Оптимизация с профилированием

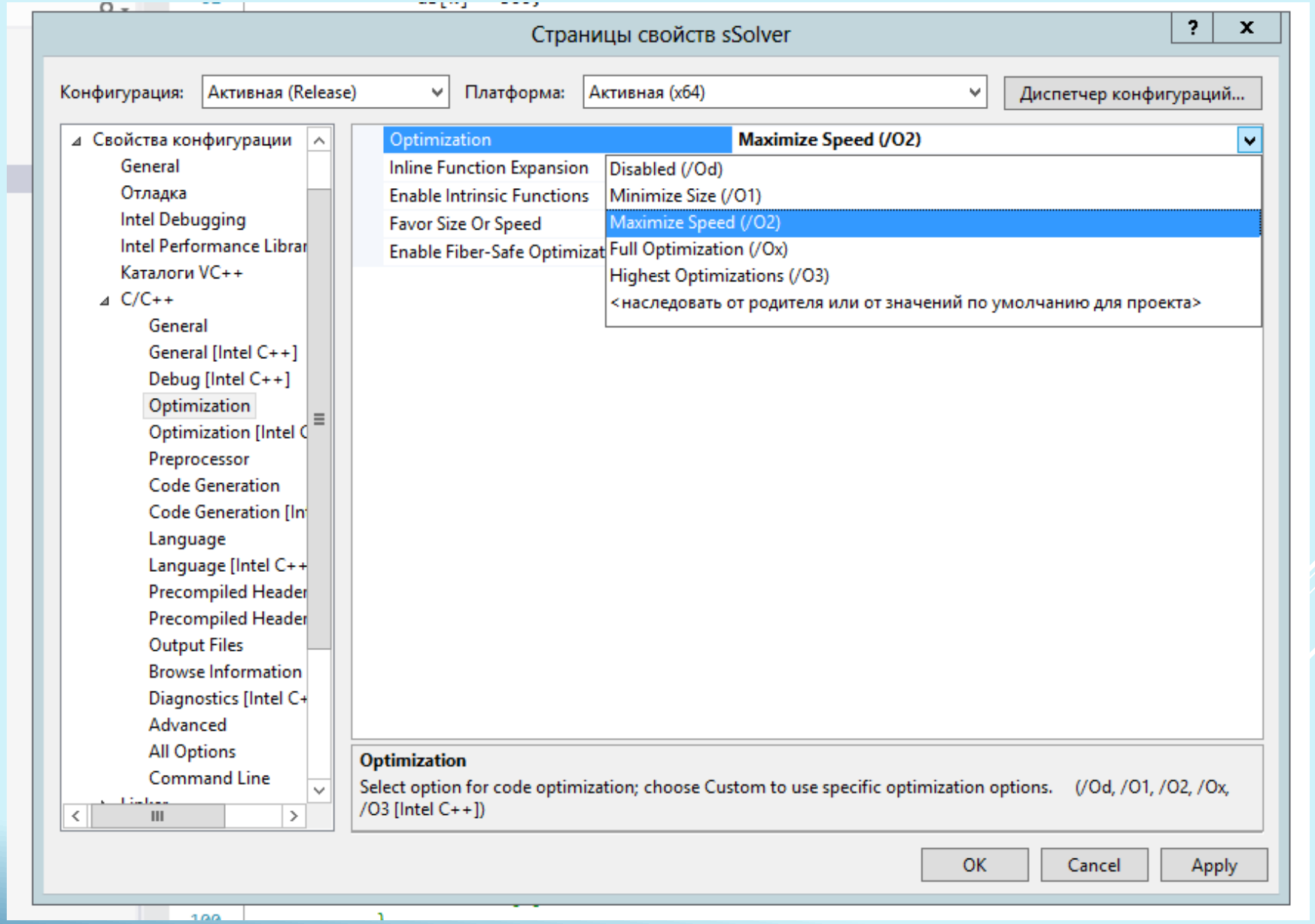
`/Qprof-gen (Windows), -prof-gen (Linux)`
`/Qprof-use (Windows), -prof-use (Linux)`

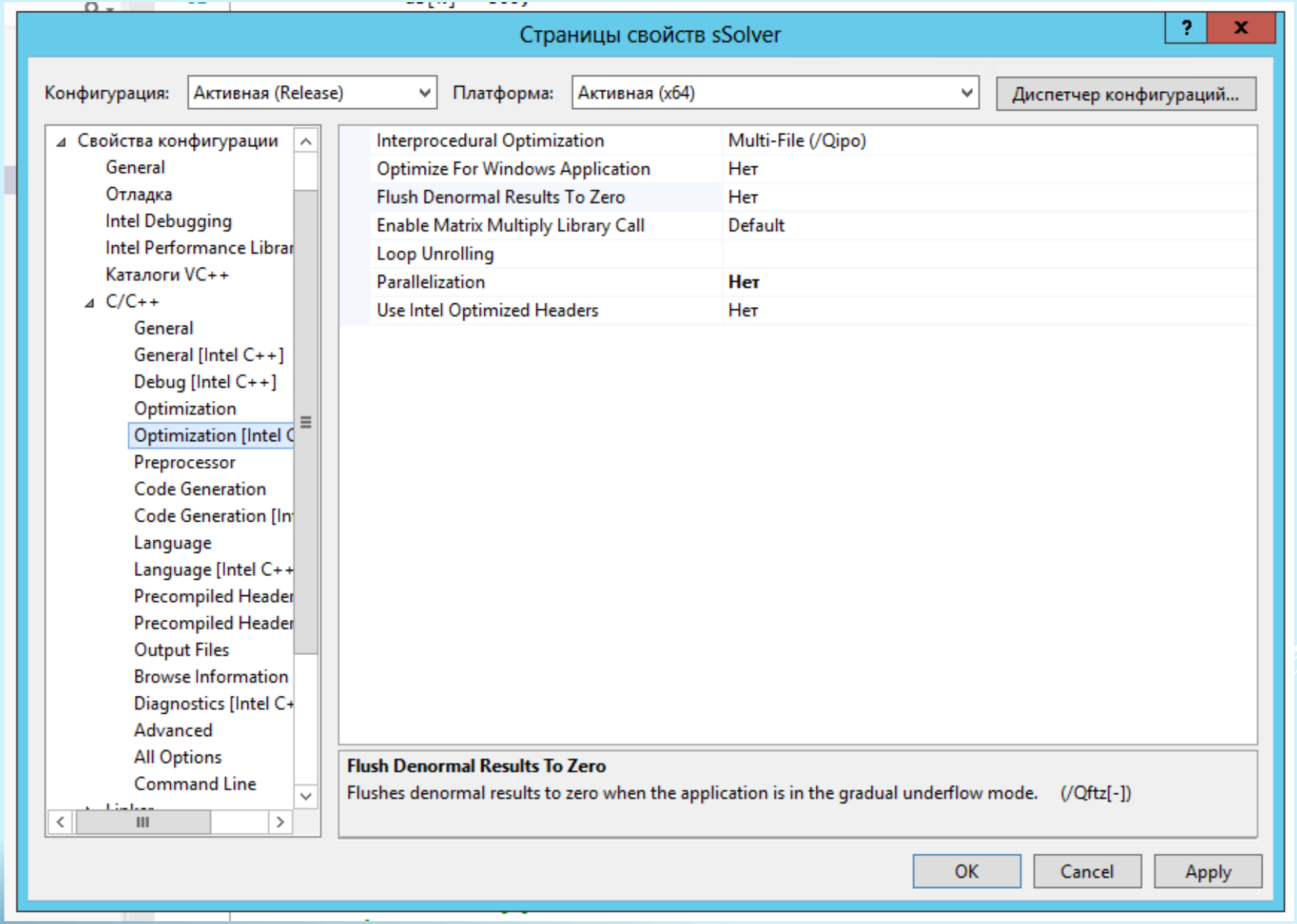
- Инструментовка.
- Сбор информации.
- Компиляция с учетом проанализированных данных.

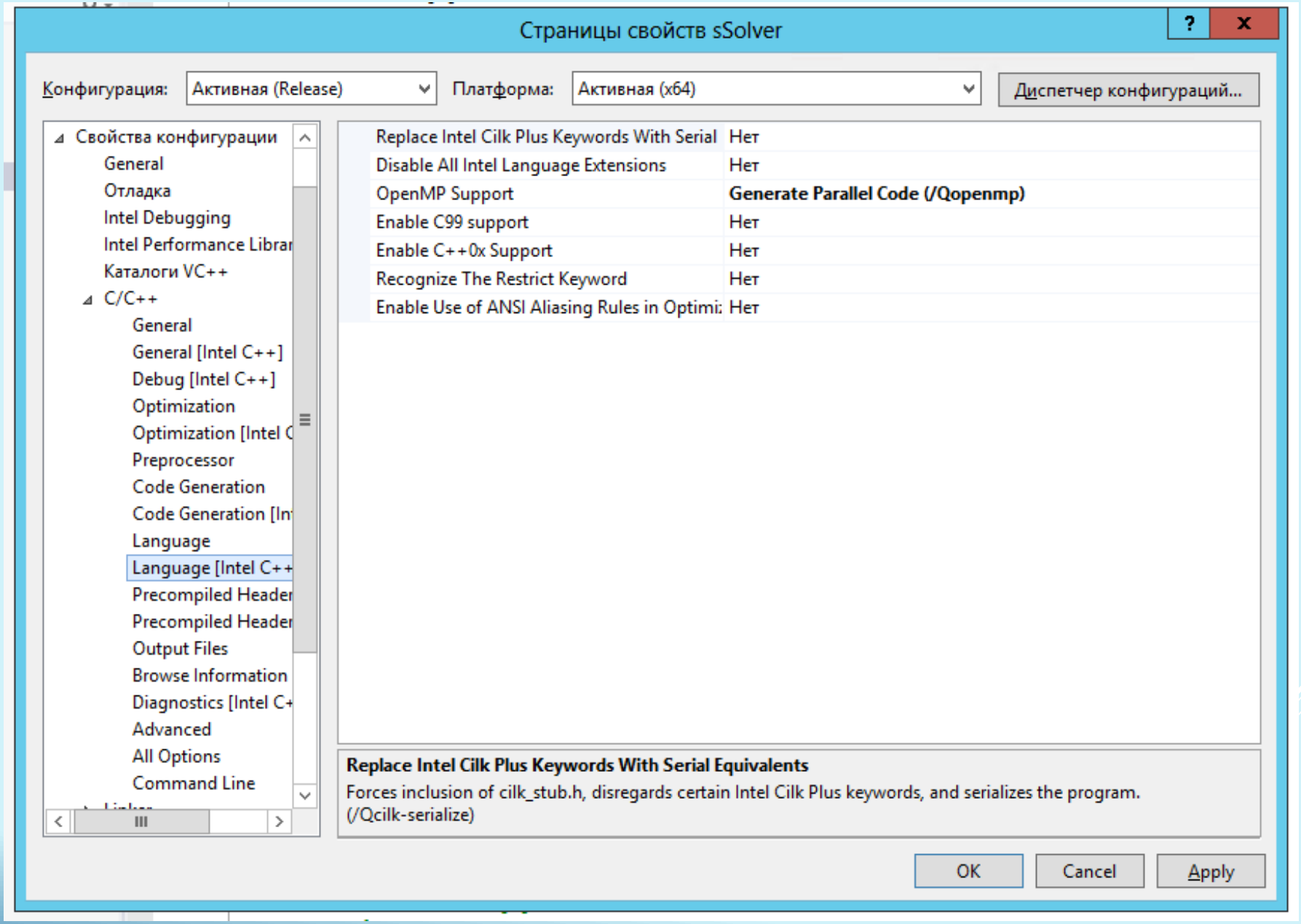
Межпроцедурная оптимизация

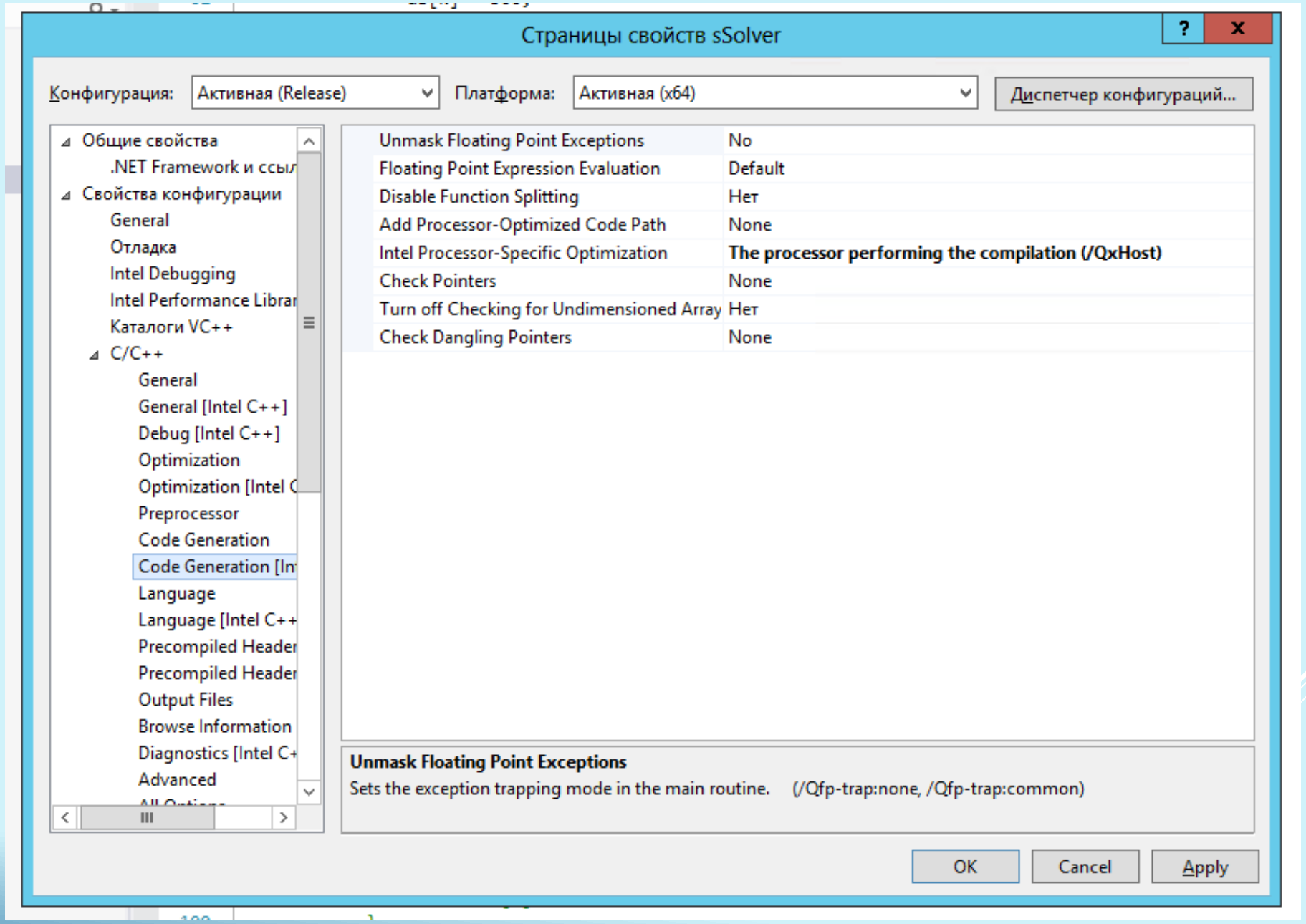
- `/Qip (Windows), -ip (Linux)`
- Анализ вызываемых функций приложения.
- Оптимизация многочисленных «маленьких» функций, особенно в циклах.
- Встраивание (inlining, подстановка кода) – уменьшение накладных расходов, создание возможностей для других видов оптимизации.
- Удаление неиспользуемого кода.
- Замена виртуальных вызовов статическими.
- Замена параметра функции константой.
- Эффективный анализ зависимостей для оптимизации циклов, векторизации и распараллеливания.
- Размер бинарного файла и время компиляции увеличиваются.

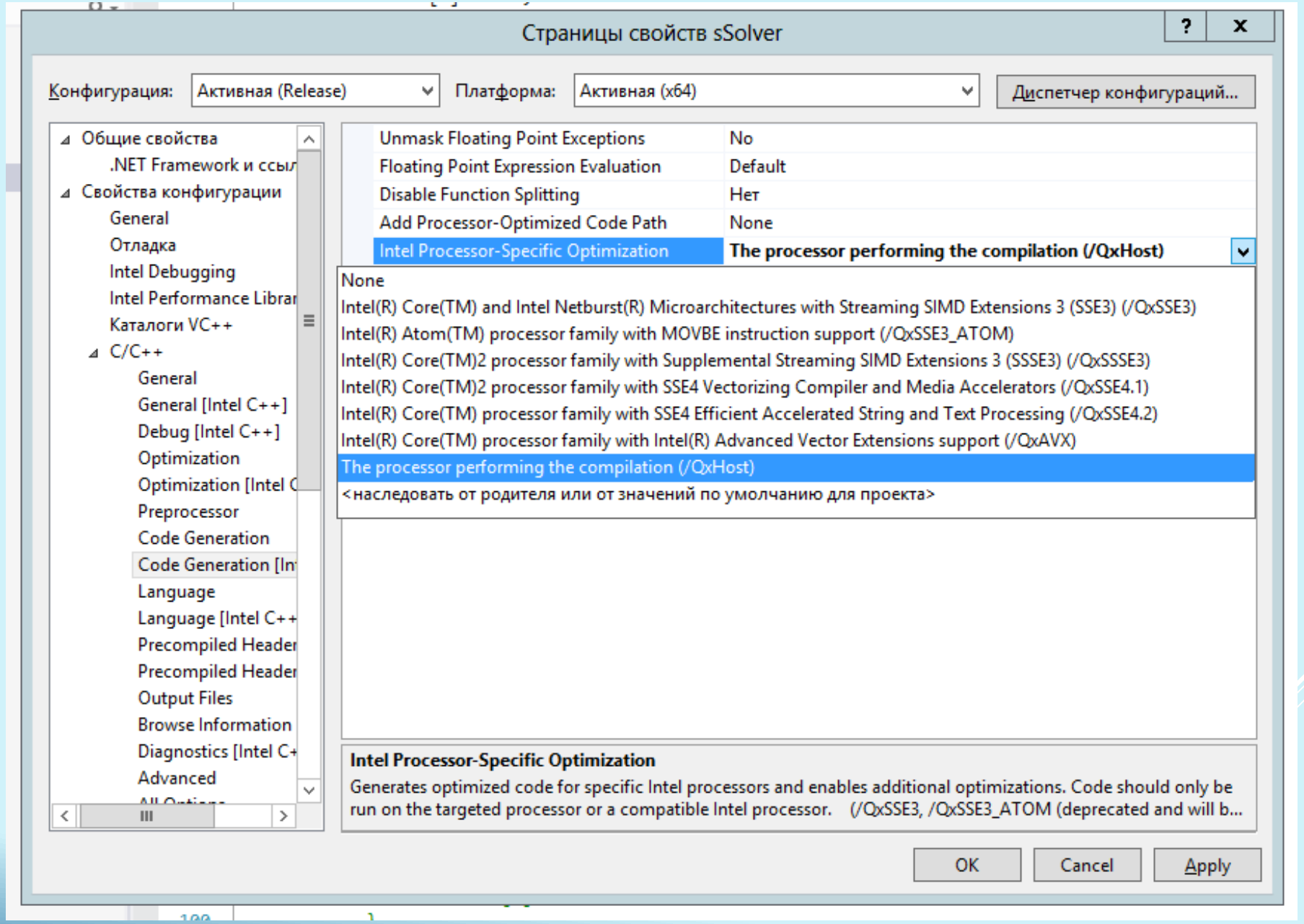

61 ab[k] = 300;
Страницы свойств sSolver
Конфигурация: Активная (Release)
Платформа: Активная (x64)
Диспетчер конфигураций...
Свойства конфигурации
General
Отладка
Intel Debugging
Intel Performance Librar
Каталоги VC++
C/C++
General
General [Intel C++]
Debug [Intel C++]
Optimization
Optimization [Intel C
Preprocessor
Code Generation
Code Generation [In
Language
Language [Intel C++
Precompiled Header
Precompiled Header
Output Files
Browse Information
Diagnostics [Intel C+
Advanced
All Options
Command Line
Optimization Maximize Speed (/O2)
Inline Function Expansion Default
Enable Intrinsic Functions Да (/Oi)
Favor Size Or Speed Favor fast code (/Ot)
Enable Fiber-Safe Optimizations Нет
Optimization
Select option for code optimization; choose Custom to use specific optimization options. (/Od, /O1, /O2, /Ox, /O3 [Intel C++])
OK
Cancel
Apply
100 }
101 fvin.close();

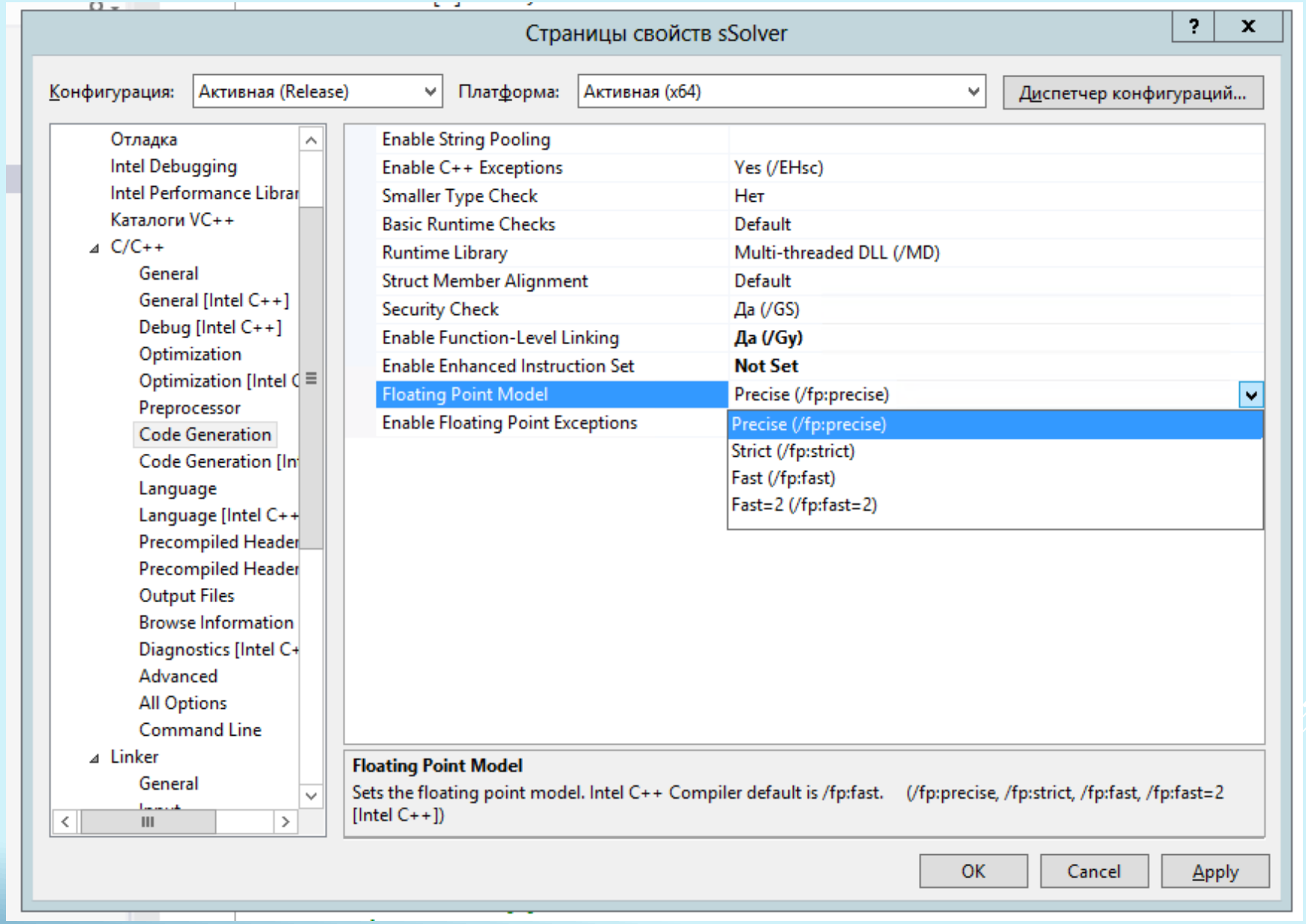

Страницы свойств sSolver
Конфигурация: Активная (Release)
Платформа: Активная (x64)
Диспетчер конфигураций...
Свойства конфигурации
General
Отладка
Intel Debugging
Intel Performance Librar
Каталоги VC++
C/C++
General
General [Intel C++]
Debug [Intel C++]
Optimization
Optimization [Intel C
Preprocessor
Code Generation
Code Generation [In
Language
Language [Intel C++
Precompiled Header
Precompiled Header
Output Files
Browse Information
Diagnostics [Intel C+
Advanced
All Options
Command Line
Optimization
Maximize Speed (/O2)
Inline Function Expansion
Enable Intrinsic Functions
Favor Size Or Speed
Enable Fiber-Safe Optimizat
Disabled (/Od)
Minimize Size (/O1)
Maximize Speed (/O2)
Full Optimization (/Ox)
Highest Optimizations (/O3)
<наследовать от родителя или от значений по умолчанию для проекта>
Optimization
Select option for code optimization; choose Custom to use specific optimization options. (/Od, /O1, /O2, /Ox, /O3 [Intel C++])
OK
Cancel
Apply

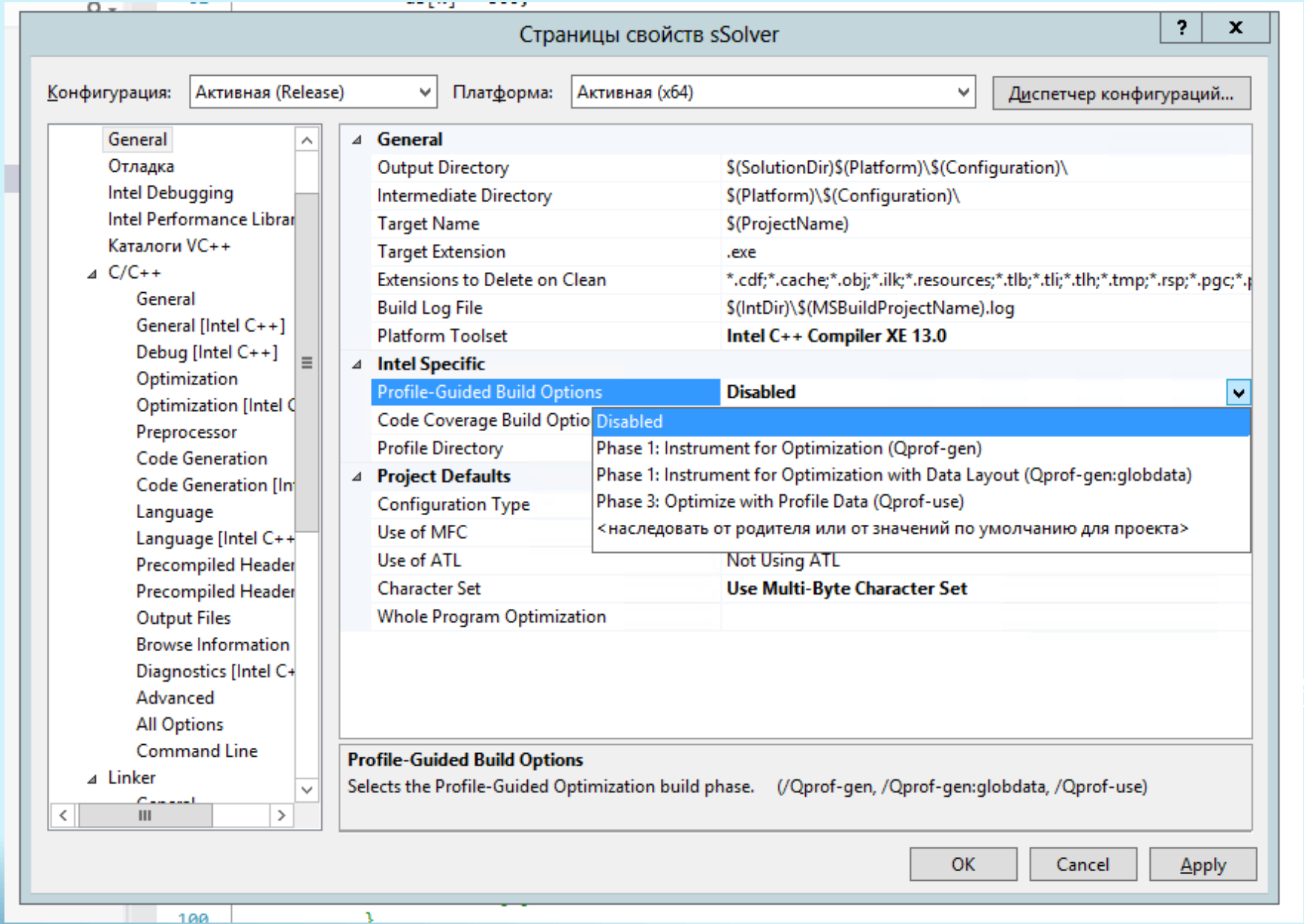

Страницы свойств sSolver
Конфигурация: Активная (Release)
Платформа: Активная (x64)
Диспетчер конфигураций...
Свойства конфигурации
General
Отладка
Intel Debugging
Intel Performance Librar
Каталоги VC++
C/C++
General
General [Intel C++]
Debug [Intel C++]
Optimization
Optimization [Intel C
Preprocessor
Code Generation
Code Generation [In
Language
Language [Intel C++
Precompiled Header
Precompiled Header
Output Files
Browse Information
Diagnostics [Intel C+
Advanced
All Options
Command Line
Interprocedural Optimization Multi-File (/Qipo)
Optimize For Windows Application Нет
Flush Denormal Results To Zero Нет
Enable Matrix Multiply Library Call Default
Loop Unrolling
Parallelization Нет
Use Intel Optimized Headers Нет
Flush Denormal Results To Zero
Flushes denormal results to zero when the application is in the gradual underflow mode. (/Qftz[-])
OK
Cancel
Apply

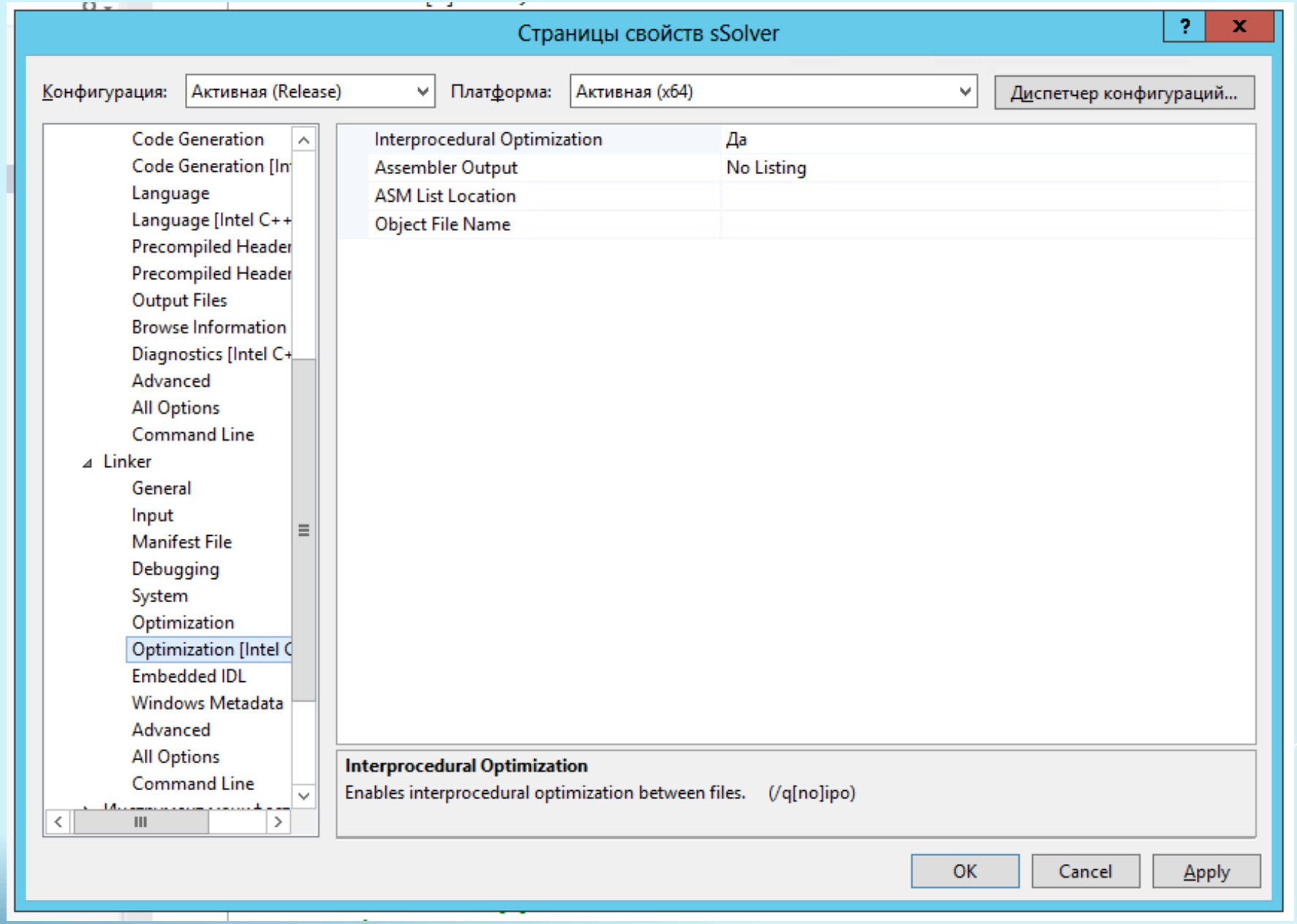

Страницы свойств sSolver
Конфигурация: Активная (Release)
Платформа: Активная (x64)
Диспетчер конфигураций...
Свойства конфигурации
General
Отладка
Intel Debugging
Intel Performance Librar
Каталоги VC++
C/C++
General
General [Intel C++]
Debug [Intel C++]
Optimization
Optimization [Intel C
Preprocessor
Code Generation
Code Generation [In
Language
Language [Intel C++
Precompiled Header
Precompiled Header
Output Files
Browse Information
Diagnostics [Intel C+
Advanced
All Options
Command Line
Replace Intel Cilk Plus Keywords With Serial Нет
Disable All Intel Language Extensions Нет
OpenMP Support Generate Parallel Code (/Qopenmp)
Enable C99 support Нет
Enable C++0x Support Нет
Recognize The Restrict Keyword Нет
Enable Use of ANSI Aliasing Rules in Optimi: Нет
Replace Intel Cilk Plus Keywords With Serial Equivalents
Forces inclusion of cilk_stub.h, disregards certain Intel Cilk Plus keywords, and serializes the program.
(/Qcilk-serialize)
OK
Cancel
Apply


Страницы свойств sSolver
Конфигурация: Активная (Release)
Платформа: Активная (x64)
Диспетчер конфигураций...
Общие свойства
.NET Framework и ссыл
Свойства конфигурации
General
Отладка
Intel Debugging
Intel Performance Librar
Каталоги VC++
C/C++
General
General [Intel C++]
Debug [Intel C++]
Optimization
Optimization [Intel C
Preprocessor
Code Generation
Code Generation [In
Language
Language [Intel C++
Precompiled Header
Precompiled Header
Output Files
Browse Information
Diagnostics [Intel C+
Advanced
Unmask Floating Point Exceptions No
Floating Point Expression Evaluation Default
Disable Function Splitting Нет
Add Processor-Optimized Code Path None
Intel Processor-Specific Optimization The processor performing the compilation (/QxHost)
Check Pointers None
Turn off Checking for Undimensioned Array Нет
Check Dangling Pointers None
Unmask Floating Point Exceptions
Sets the exception trapping mode in the main routine. (/Qfp-trap:none, /Qfp-trap:common)
OK
Cancel
Apply


Страницы свойств sSolver
Конфигурация: Активная (Release)
Платформа: Активная (x64)
Диспетчер конфигураций...
Общие свойства
.NET Framework и ссыл
Свойства конфигурации
General
Отладка
Intel Debugging
Intel Performance Librar
Каталоги VC++
C/C++
General
General [Intel C++]
Debug [Intel C++]
Optimization
Optimization [Intel C
Preprocessor
Code Generation
Code Generation [In
Language
Language [Intel C++
Precompiled Header
Precompiled Header
Output Files
Browse Information
Diagnostics [Intel C+
Advanced
Unmask Floating Point Exceptions
No
Floating Point Expression Evaluation
Default
Disable Function Splitting
Нет
Add Processor-Optimized Code Path
None
Intel Processor-Specific Optimization
The processor performing the compilation (/QxHost)
None
Intel(R) Core(TM) and Intel Netburst(R) Microarchitectures with Streaming SIMD Extensions 3 (SSE3) (/QxSSE3)
Intel(R) Atom(TM) processor family with MOVBE instruction support (/QxSSE3_ATOM)
Intel(R) Core(TM)2 processor family with Supplemental Streaming SIMD Extensions 3 (SSSE3) (/QxSSSE3)
Intel(R) Core(TM)2 processor family with SSE4 Vectorizing Compiler and Media Accelerators (/QxSSE4.1)
Intel(R) Core(TM) processor family with SSE4 Efficient Accelerated String and Text Processing (/QxSSE4.2)
Intel(R) Core(TM) processor family with Intel(R) Advanced Vector Extensions support (/QxAVX)
The processor performing the compilation (/QxHost)
<наследовать от родителя или от значений по умолчанию для проекта>
Intel Processor-Specific Optimization
Generates optimized code for specific Intel processors and enables additional optimizations. Code should only be run on the targeted processor or a compatible Intel processor. (/QxSSE3, /QxSSE3_ATOM (deprecated and will b...
OK
Cancel
Apply


Страницы свойств sSolver
Конфигурация: Активная (Release)
Платформа: Активная (x64)
Диспетчер конфигураций...
Отладка
Intel Debugging
Intel Performance Librar
Каталоги VC++
C/C++
General
General [Intel C++]
Debug [Intel C++]
Optimization
Optimization [Intel C
Preprocessor
Code Generation
Code Generation [In
Language
Language [Intel C++
Precompiled Header
Precompiled Header
Output Files
Browse Information
Diagnostics [Intel C+
Advanced
All Options
Command Line
Linker
General
Enable String Pooling
Enable C++ Exceptions Yes (/EHsc)
Smaller Type Check Нет
Basic Runtime Checks Default
Runtime Library Multi-threaded DLL (/MD)
Struct Member Alignment Default
Security Check Да (/GS)
Enable Function-Level Linking Да (/Gy)
Enable Enhanced Instruction Set Not Set
Floating Point Model Precise (/fp:precise)
Enable Floating Point Exceptions
Precise (/fp:precise)
Strict (/fp:strict)
Fast (/fp:fast)
Fast=2 (/fp:fast=2)
Floating Point Model
Sets the floating point model. Intel C++ Compiler default is /fp:fast. (/fp:precise, /fp:strict, /fp:fast, /fp:fast=2 [Intel C++])
OK
Cancel
Apply


Страницы свойств sSolver
Конфигурация:
Активная (Release)
Платформа:
Активная (x64)
Диспетчер конфигураций...
General
Отладка
Intel Debugging
Intel Performance Librar
Каталоги VC++
C/C++
General
General [Intel C++]
Debug [Intel C++]
Optimization
Optimization [Intel C
Preprocessor
Code Generation
Code Generation [In
Language
Language [Intel C++
Precompiled Header
Precompiled Header
Output Files
Browse Information
Diagnostics [Intel C+
Advanced
All Options
Command Line
Linker
General
Output Directory
$(SolutionDir)$(Platform)\$(Configuration)\
Intermediate Directory
$(Platform)\$(Configuration)\
Target Name
$(ProjectName)
Target Extension
.exe
Extensions to Delete on Clean
.cdf;.cache;*.obj;*.ilk;*.resources;*.tlb;*.tli;*.tlh;*.tmp;*.rsp;*.pgc;*.
Build Log File
$(IntDir)\$(MSBuildProjectName).log
Platform Toolset
Intel C++ Compiler XE 13.0
Intel Specific
Profile-Guided Build Options
Disabled
Code Coverage Build Optio
Profile Directory
Project Defaults
Configuration Type
Use of MFC
Use of ATL
Not Using ATL
Character Set
Use Multi-Byte Character Set
Whole Program Optimization
Disabled
Phase 1: Instrument for Optimization (Qprof-gen)
Phase 1: Instrument for Optimization with Data Layout (Qprof-gen:globdata)
Phase 3: Optimize with Profile Data (Qprof-use)
<наследовать от родителя или от значений по умолчанию для проекта>
Profile-Guided Build Options
Selects the Profile-Guided Optimization build phase. (/Qprof-gen, /Qprof-gen:globdata, /Qprof-use)
OK
Cancel
Apply


Страницы свойств sSolver
Конфигурация:
Активная (Release)
Платформа:
Активная (x64)
Диспетчер конфигураций...
Code Generation
Code Generation [In
Language
Language [Intel C++
Precompiled Header
Precompiled Header
Output Files
Browse Information
Diagnostics [Intel C+
Advanced
All Options
Command Line
Linker
General
Input
Manifest File
Debugging
System
Optimization
Optimization [Intel C
Embedded IDL
Windows Metadata
Advanced
All Options
Command Line
Interprocedural Optimization
Да
Assembler Output
No Listing
ASM List Location
Object File Name
Interprocedural Optimization
Enables interprocedural optimization between files. (/q[no]ipo)
OK
Cancel
Apply

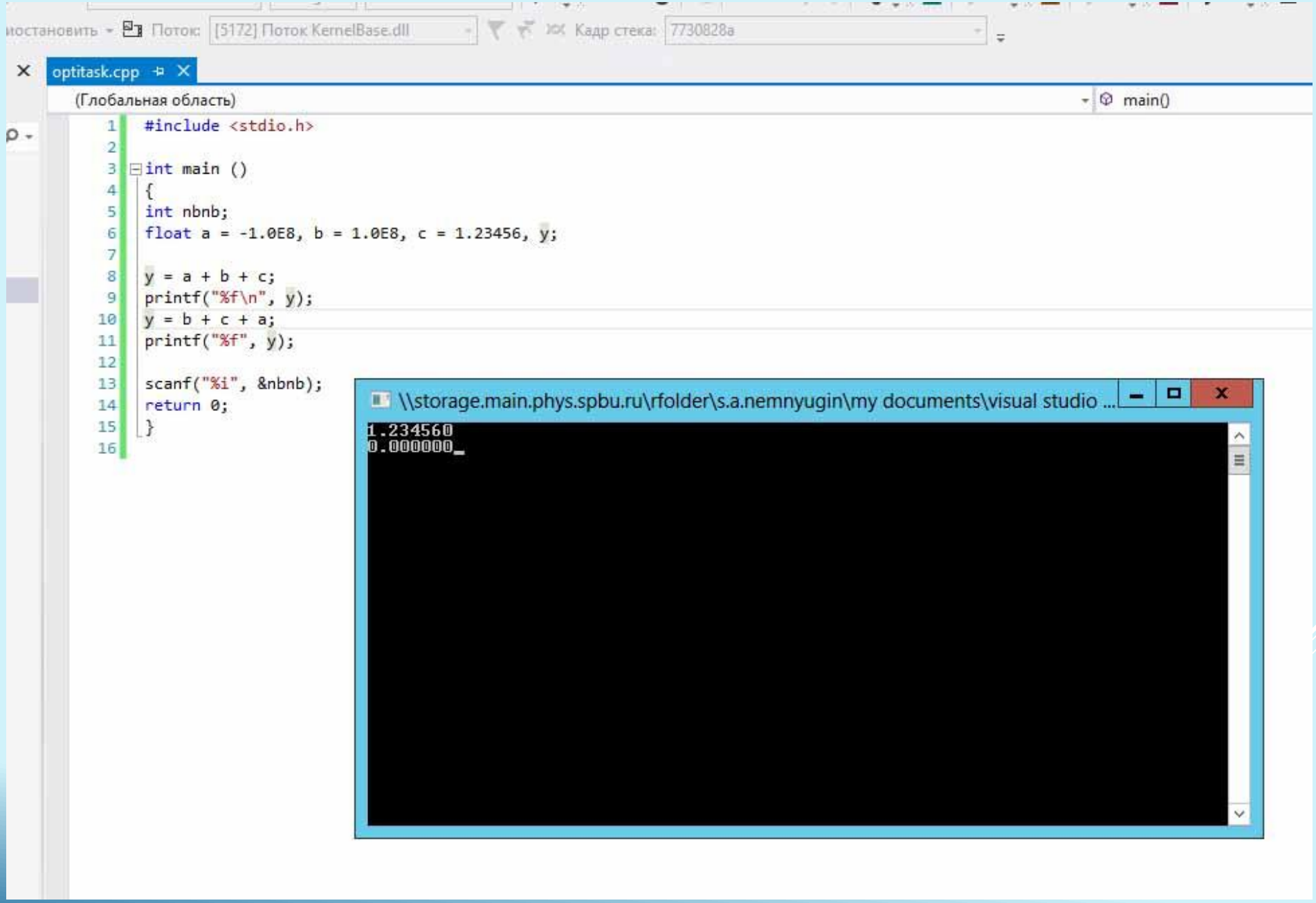
Изменение последовательности выполнения операций

При выполнении арифметических операций с плавающей точкой изменение последовательности операндов может привести к изменению результата вычисления арифметического выражения. В примере:

```
float a = -1.0E8, b = 1.0E8, c = 1.23456, y;
y = a + b + c;
printf("%f\n", y);
 y = b + c + a;
printf("%f", y);
```

оба результата – разные.

…


остановить
Поток:
[5172] Поток KernelBase.dll
Кадр стека:
7730828a
optitask.cpp
(Глобальная область)
main()
#include <stdio.h>
int main ()
{
int nbnb;
float a = -1.0E8, b = 1.0E8, c = 1.23456, y;
y = a + b + c;
printf("%f\n", y);
y = b + c + a;
printf("%f", y);
scanf("%i", &nbnb);
return 0;
}
\\storage.main.phys.spbu.ru\rfolder\s.a.nemnyugin\my documents\visual studio ...
1.234560
0.000000