Help: настройка Visual Studio.Net для создания консоль-приложения на основе Intel C++ с применением OpenMP. Инструменты

«Практическое параллельное программирование в системах с общей памятью»

Содержание

- 1. Последовательность установки программ
- 2.Создание консоль-приложения на C++ с применением OpenMP (Microsoft C++, Intel C++)
- 3.Intel Thread Checker (тестирование правильности выполнения многопоточного приложения)
- 4.Intel Thread Profiler (тестирование производительности многопоточных вычислений)

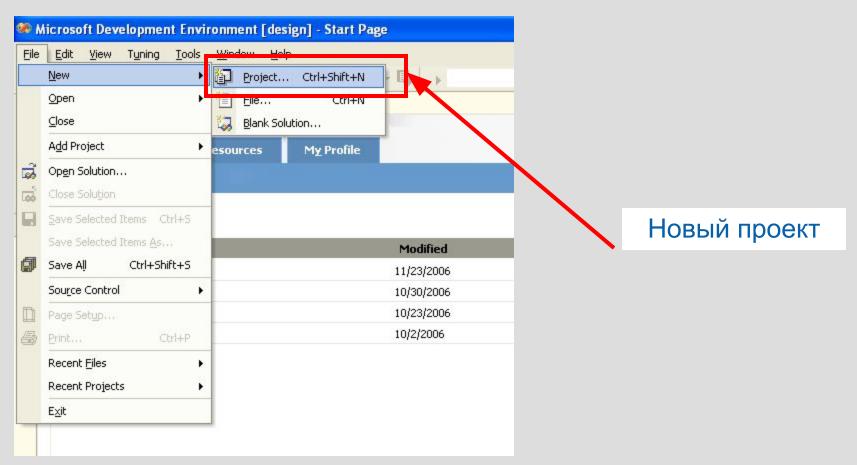
Последовательность установки

- 1. Visual Studio. Net
- 2.Intel C++
- 3.Intel VTune Performance Analyzer
- 4.Intel Thread Checker или Intel Thread Profiler

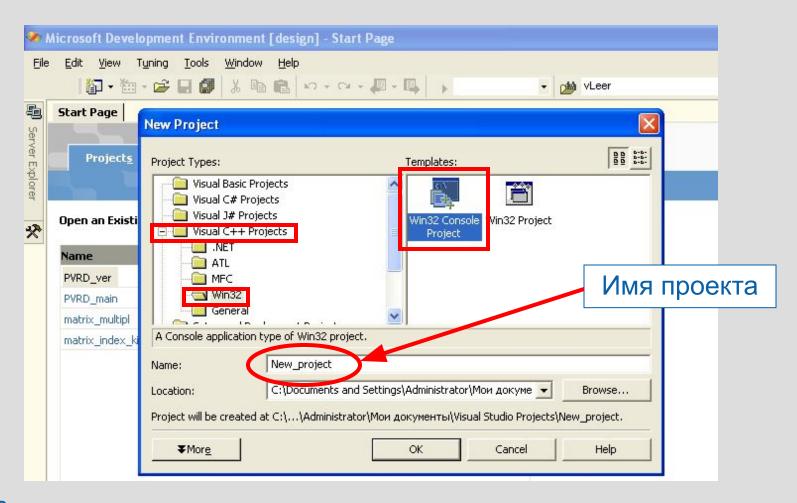
2. Создание консоль-приложения на C++ с применением OpenMP (Microsoft C++, Intel C++)

- 2.1. Создание консоль-приложения на основе С++
- 2.2. Преобразование в проект на основе Intel C++
- 2.3. Установка поддержки директив OpenMP
- 2.3. Настройка на многопоточно-безопасные библиотеки

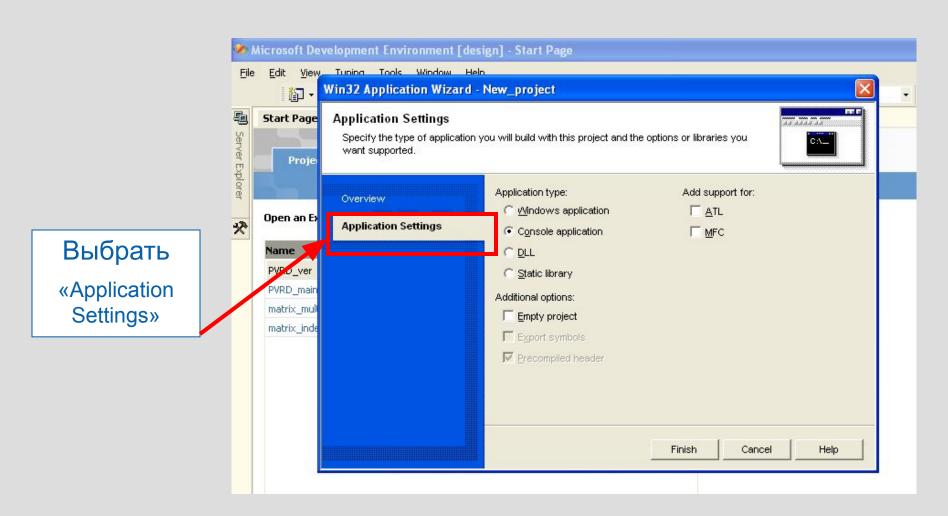
2.1. Создание консоль-приложения на С++



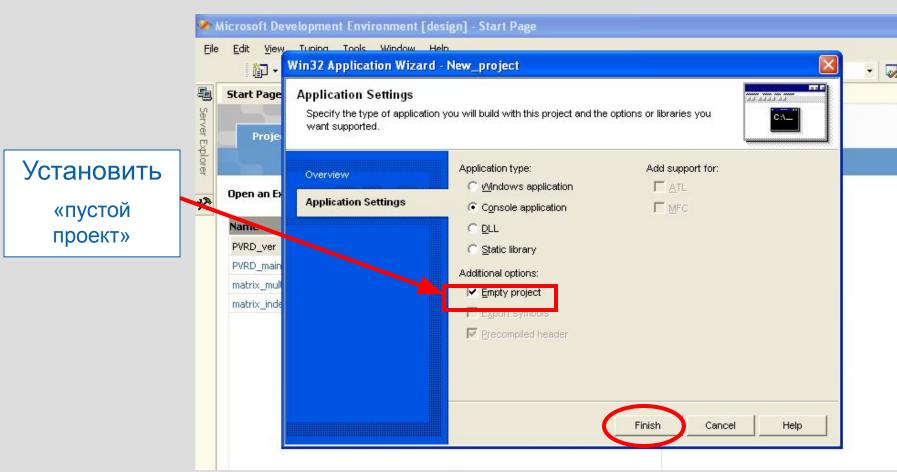
2.1. Создание консоль-приложения на C++ (слайд 2)



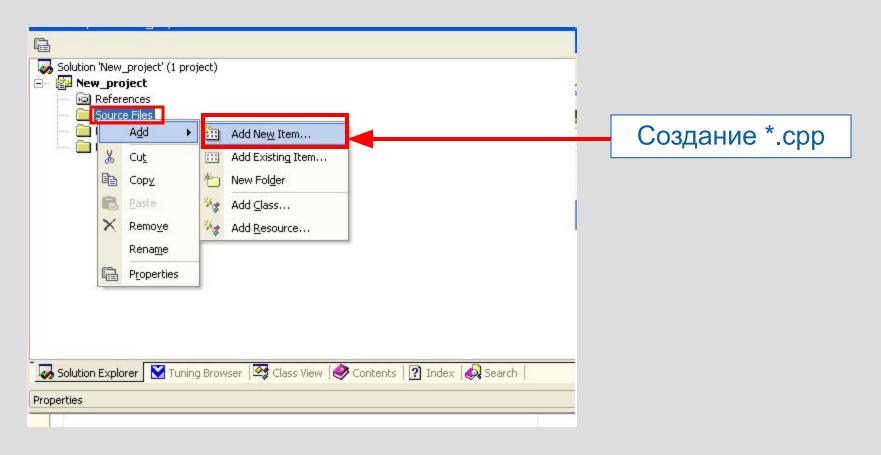
2.1. Создание консоль-приложения на C++ (слайд 3)



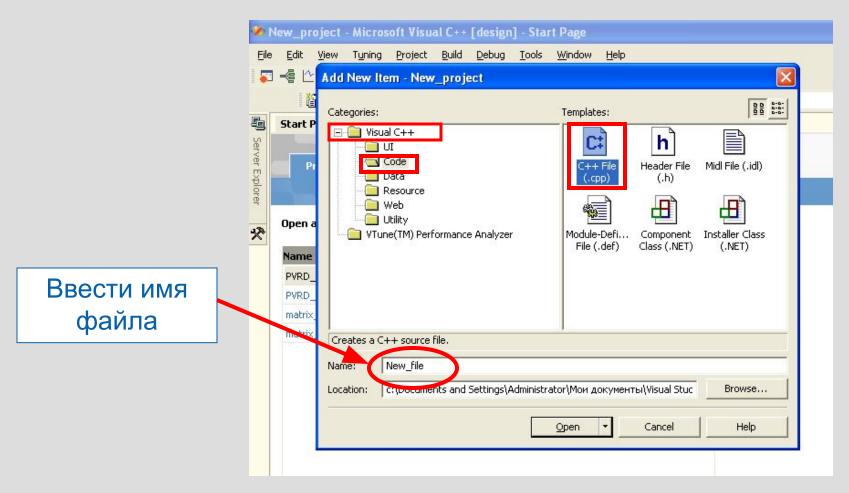
2.1. Создание консоль-приложения на C++ (слайд 4)



2.1. Создание консоль-приложения на C++ (слайд 5)

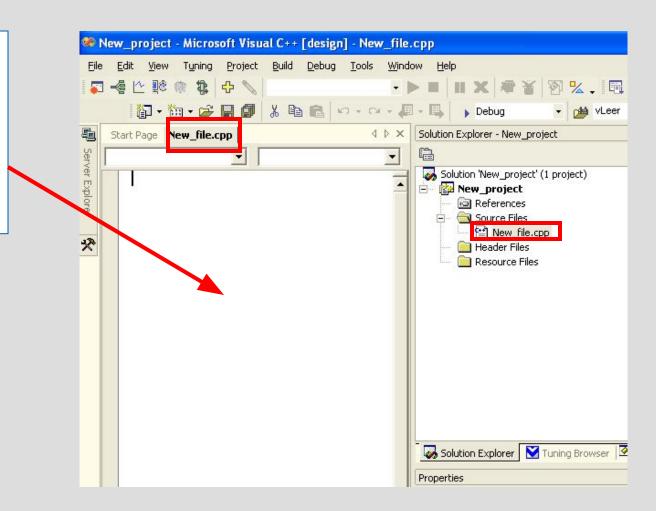


2.1. Создание консоль-приложения на C++ (слайд 6)



2.1. Создание консоль-приложения на C++ (слайд 7)

В открывшемся окне набрать текст новой С++ - программы или скопировать в это окно текст уже имеющегося *.cpp

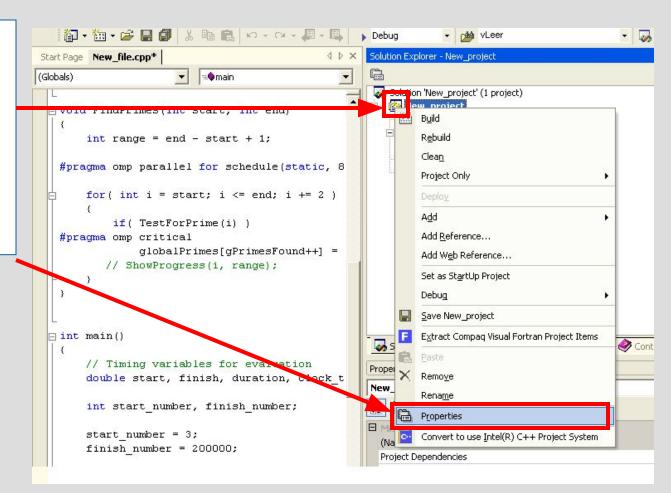


2.2. Преобразование в проект на основе Intel C++

節・猫・G □ ● X 時 高 い・ロ・Д・耳 → p vLeer Solution Explorer - New project С помощью щелчка Start Page New_file.cpp ≔**♦**main (Globals) Solation 'New_project' (1 project) правой кнопки мыши на factor ++: 🕮 New project Build значке проекта открыть return (factor > limit); Rebuild Clean контекстное меню и void FindPrimes(int start, in Project Only выбрать самый нижний int range = end - start + Add #pragma omp parallel for scheпункт меню -Add Reference... for (int i = start; i <= . Add Web Reference... «Convert to use Intel(R) C++ Set as StartUp Project if (TestForPrime (i)) project System» #pragma omp critical Debug globalPrimes[gPri Save New_project // ShowProgress(i, ran-- преобразование в Extract Compaq Visual Fortran Project Items проект на основе Intel Propi X Remove New □ int main() Rename C++Properties // Timing variables for e double start, finish, dur Convert to use Intel(R) C++ Project System int start number, finish : Project File

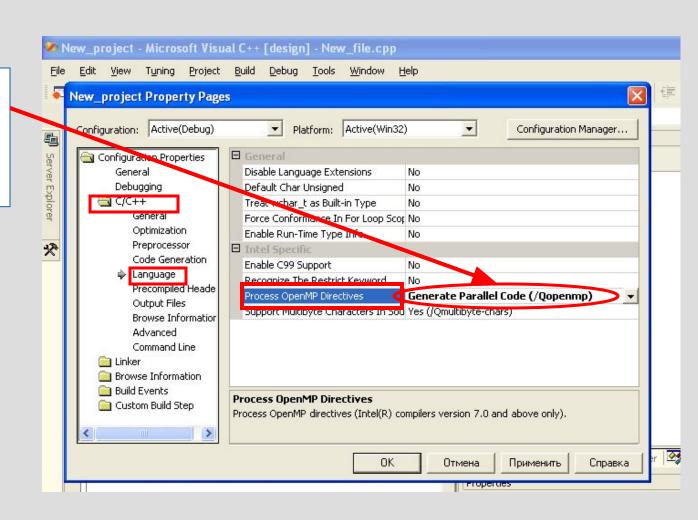
2.3. Установка поддержки директив OpenMP (слайд 1)

Правым щелчком мыши на значке проекта открыть контекстное меню и выбрать окно свойств проекта

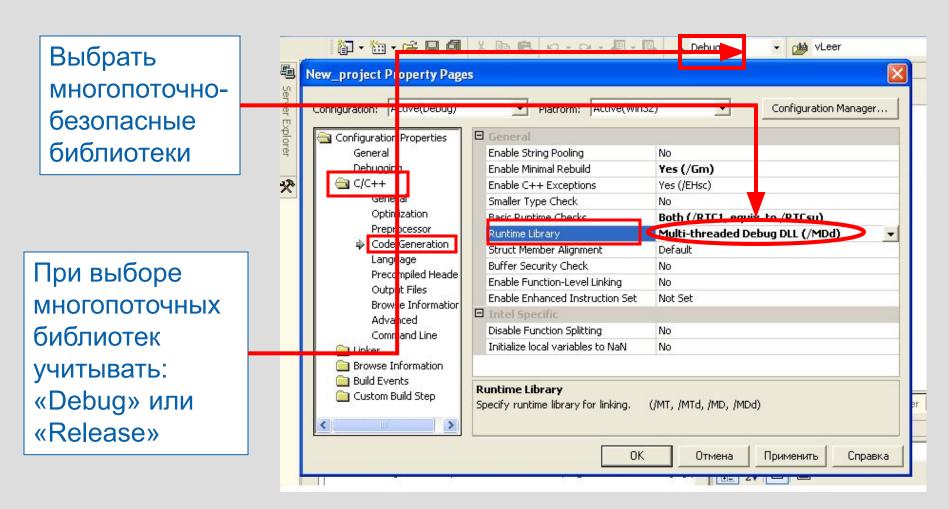


2.3. Установка поддержки директив OpenMP (слайд 2)

Установить поддержку директив ОрепМР



2.4. Настройка на многопоточнобезопасные библиотеки



3. Intel Thread Checker (тестирование правильности выполнения многопоточного приложения)

- 3.1. Подготовка приложения для анализа Thread Checker: условия компиляции
- 3.2. Настройки Visual Studio.Net для компиляции приложения с целью анализа Thread Checker
- 3.3. Выполнение анализа Thread Checker при минимальном инструментировании приложения

Дополнительная информация о Thread Checker – в лекциях 4, 8, 9

3.1. Подготовка приложения для анализа Thread Checker: условия компиляции

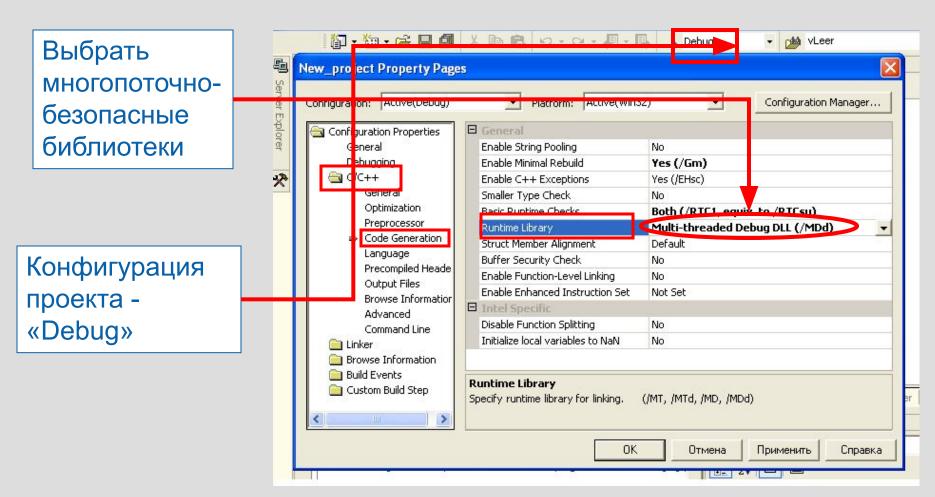
Компиляция

- Используйте многопоточно безопасные библиотеки (/MD, /MDd)
- Включите генерацию символьной информации (/zi, /zI, /z7)
- Отключите оптимизацию (/od)

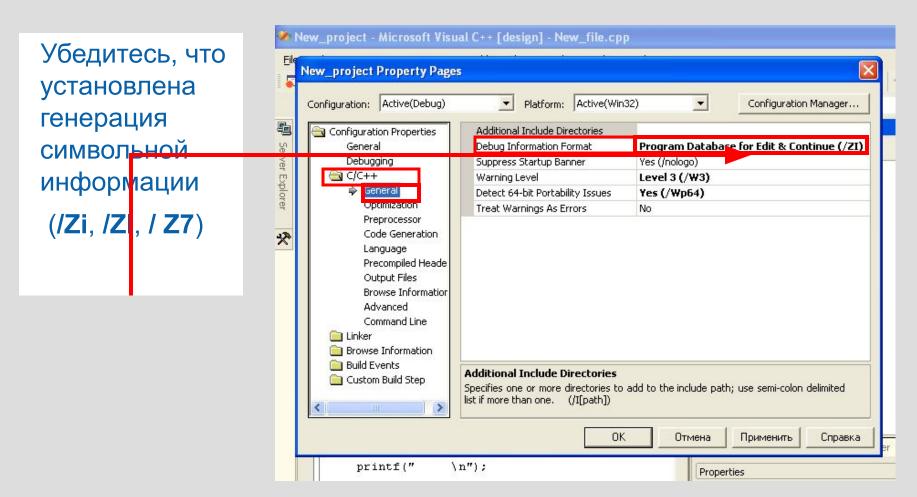
«Линкование» (Link)

- Сохранить символьную информацию (/debug)
- Specify relocatable code sections: /fixed:no)

3.2. Hастройки Visual Studio.Net для компиляции приложения с целью анализа Thread Checker (слайд 1)

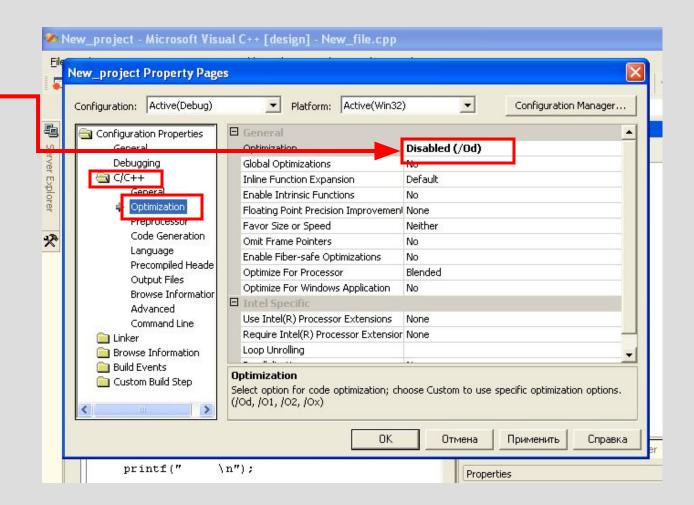


3.2. Hастройки Visual Studio.Net для компиляции приложения с целью анализа Thread Checker (слайд 2)

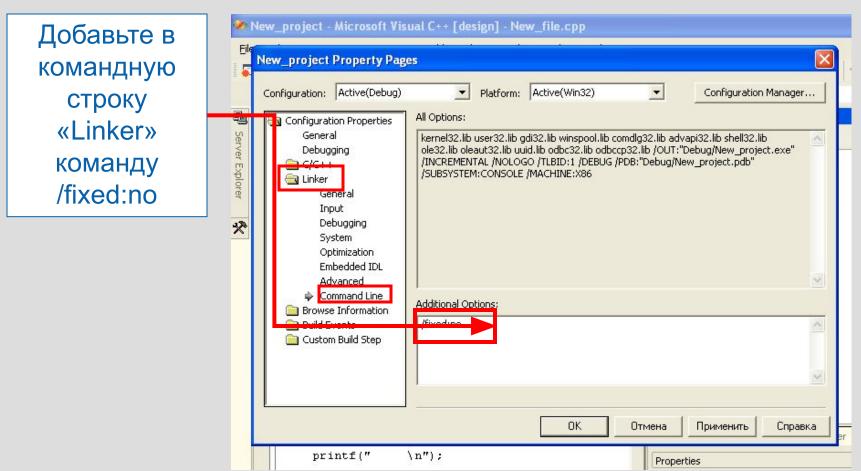


3.2. Hacтройки Visual Studio.Net для компиляции приложения с целью анализа Thread Checker (слайд 3)

Убедитесь, что отключена оптимизация



3.2. Hастройки Visual Studio.Net для компиляции приложения с целью анализа Thread Checker (слайд 4)



3.3. Выполнение анализа Thread Checker при минимальном инструментировании приложения (слайд 1)

Запуск приложения

- Должен быть выполнен из-под Thread Checker
- Приложение инструментируется во время выполнения
- Также применяются внешние инструментированные динамические библиотеки (DLLs)

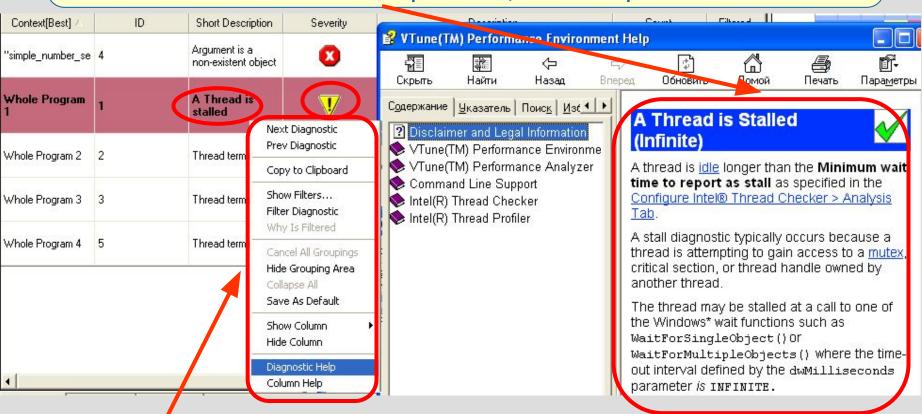
3.3. Выполнение анализа Thread Checker при минимальном инструментировании приложения (слайд 1)

Выполнение приложения

- Запуск в среде VTune™
- Запуск из-под командной строки Windows*
 - Полученные данные размещаются в файле результатов threadchecker.thr
 - Просмотр результатов (.thr file) в среде VTune

Помощь Thread Checker

Выявляет причины ошибок и предлагает способы их ликвидации. Предлагает на выбор набор API в качестве определяемых пользователем синхронизационных примитивов



Контекстное меню (щелчок правой кнопкой): выбор помощи по результатам диагностики (Diagnostic Help)

4. Intel Thread Profiler (тестирование производительности многопоточных вычислений)

4.1. Установки для выполнения минимального анализа с помощью Thread Profiler

- 4.1.1. Thread Profiler для Windows Threads
- 4.1.2. Thread Profiler для OpenMP

4.2. Немного o Thread Profiler

4.1. Установки для выполнения минимального анализа с помощью Thread Profiler

4.1.1. Thread Profiler для Windows Threads

Установки аналогичны случаю выполнения анализа с помощью Thread Checker

Дополнительная информация о Tread Profiler – в лекциях 4, 8, 9

4.1. Установки для выполнения минимального анализа с помощью Thread Profiler

4.1.2. Thread Profiler для OpenMP

• Установки аналогичны случаю выполнения анализа с помощью Thread Checker

• Дополнительная информация о Tread Profiler – в лекциях 4, 8, 9

Intel® Thread Profiler

- «Вставлен» в среду VTune™
- Сборка данных на основе работы приложения, «инструментированного» с помощью VTune
- Предназначен для анализа производительности OpenMP* приложений или многопоточных приложений с использованием потоков Win32* API и POSIX*
- Выявляет в многопоточном приложении «узкие места», которые непосредственно влияют на производительность

Как и что можно анализировать с помощью Intel® Thread Profiler

Поддерживает несколько различных компиляторов

- Компиляторы Intel® C++ и Fortran, v7 и выше
- Microsoft* Visual* C++, v6
- Microsoft* Visual* C++ .NET* 2002 & 2003 Editions
 - Интегрируется в среду Microsoft Visual Studio .NET*

Бинарное «инструментирование» приложений

Различные способы фильтрации данных и различные диаграммы для их представления с целью организации анализа

Анализ «критического пути» (critical path)