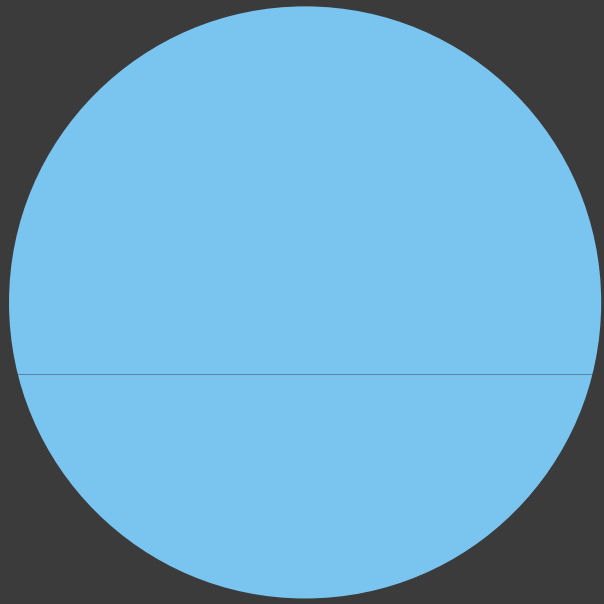
Как поставить программе правильный диагноз

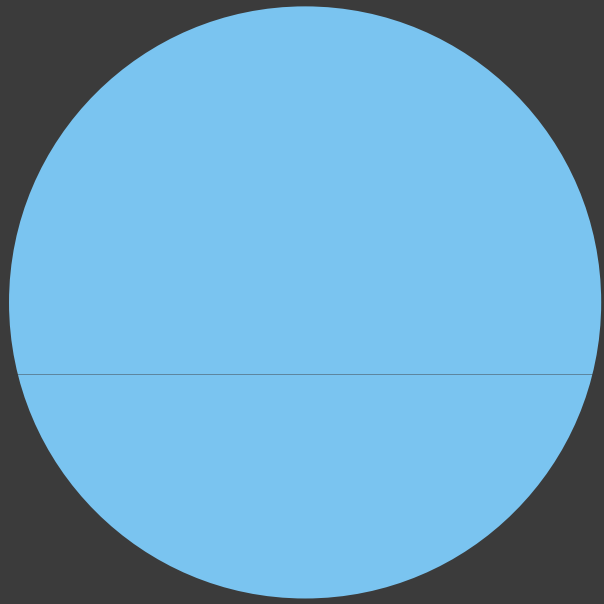


С.А.НЕМНЮГИН
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
2013

Всероссийская молодёжная школа «АТОМОСФЕРА: встраиваемые системы и робототехника – 2013»

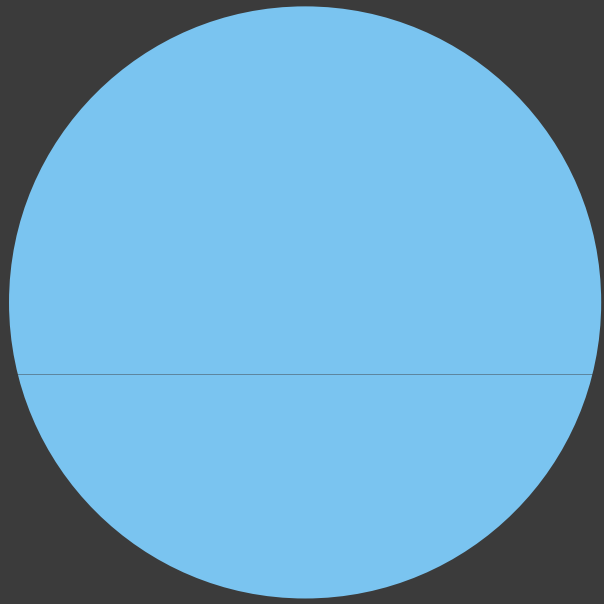
Цикл оптимизации

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.



Три уровня оптимизации:

1. Системный уровень.
2. Уровень приложения.
3. Микроархитектурный уровень.



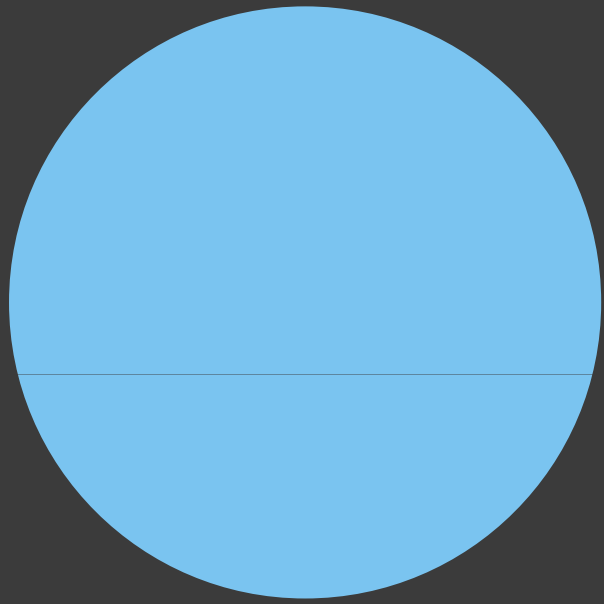
1 шаг анализа производительности программы - **на уровне системы**.

- Анализ операций обмена с внешними носителями (жесткими дисками).
- Анализ использования оперативной памяти.
- Анализ взаимодействия с сетью.

Цель оптимизации на уровне системы – добиться наиболее эффективного взаимодействия программы с системой.

Наличие проблем на уровне системы может лишить смысла оптимизацию программы на других уровнях (если программа мало загружает процессор из-за проблем на уровне системы, даже серьезное ускорение на уровне архитектуры не даст, скорее всего, заметного выигрыша).

Оптимизация **на уровне приложения** - оптимизация алгоритмов.

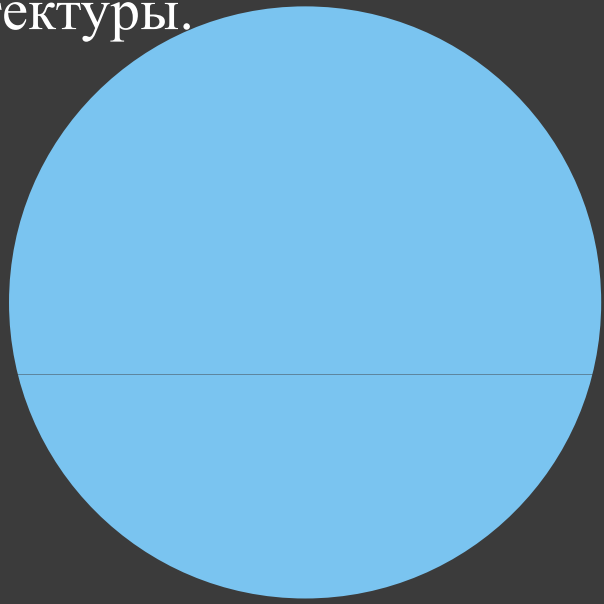


Выполняется анализ:

- эффективности использования ППИ;
- эффективности реализации многопоточности;
- наличия блокировок;
- др

3 (самый нижний) уровень анализа – уровень микроархитектуры.

Обладает наименьшим потенциалом эффективности.



Ключевые факторы:

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

Наибольший выигрыш (в несколько раз по производительности) достигается на уровне системы.

Умеренный выигрыш - на уровне приложения.

Небольшой выигрыш (до нескольких десятков процентов) - на уровне архитектуры процессора.

При анализе выполнения программы в первую очередь следует обратить внимание на следующие факторы:

- ***Количество промахов при обращении к кэш-памяти***. Каждый промах приводит к дополнительным транзакциям между оперативной памятью и кэш-памятью, что отрицательно сказывается на производительности программы.
- ***Ошибки в предсказании ветвлений***. Предсказание ветвлений позволяет прогнозировать исполнение программы, принимая меры для его ускорения. Чем точнее выполняется предсказание, тем эффективнее исполняется программа.
- ***Вычисления с плавающей точкой***. Значение этого фактора очевидно для вычислительных программ, в которых может выполняться колоссальное количество арифметических операций с плавающей точкой.

- ***Блокировки, связанные с доступом к ресурсам***. Каждая блокировка ведёт к потере производительности.
- ***Потери производительности, связанные с отсутствием выравнивания данных***. Время выполнения команд с невыровненными данными существенно увеличивается, что приводит и к увеличению времени выполнения программы.

В Help имеется информация о событиях, по которым возможен сбор статистики.

Intel® VTune Amplifier XE

Intel® VTune Amplifier XE – экс**Intel® VTune Performance Analyzer** – 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.

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 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

Позволяет выполнять анализ следующего вида:

- анализ «горячих пятен» («хотспотов»);
- анализ эффективности реализации многопоточности;
- анализ блокировок и простоев;
- анализ статистики по событиям.

Графический интерфейс:

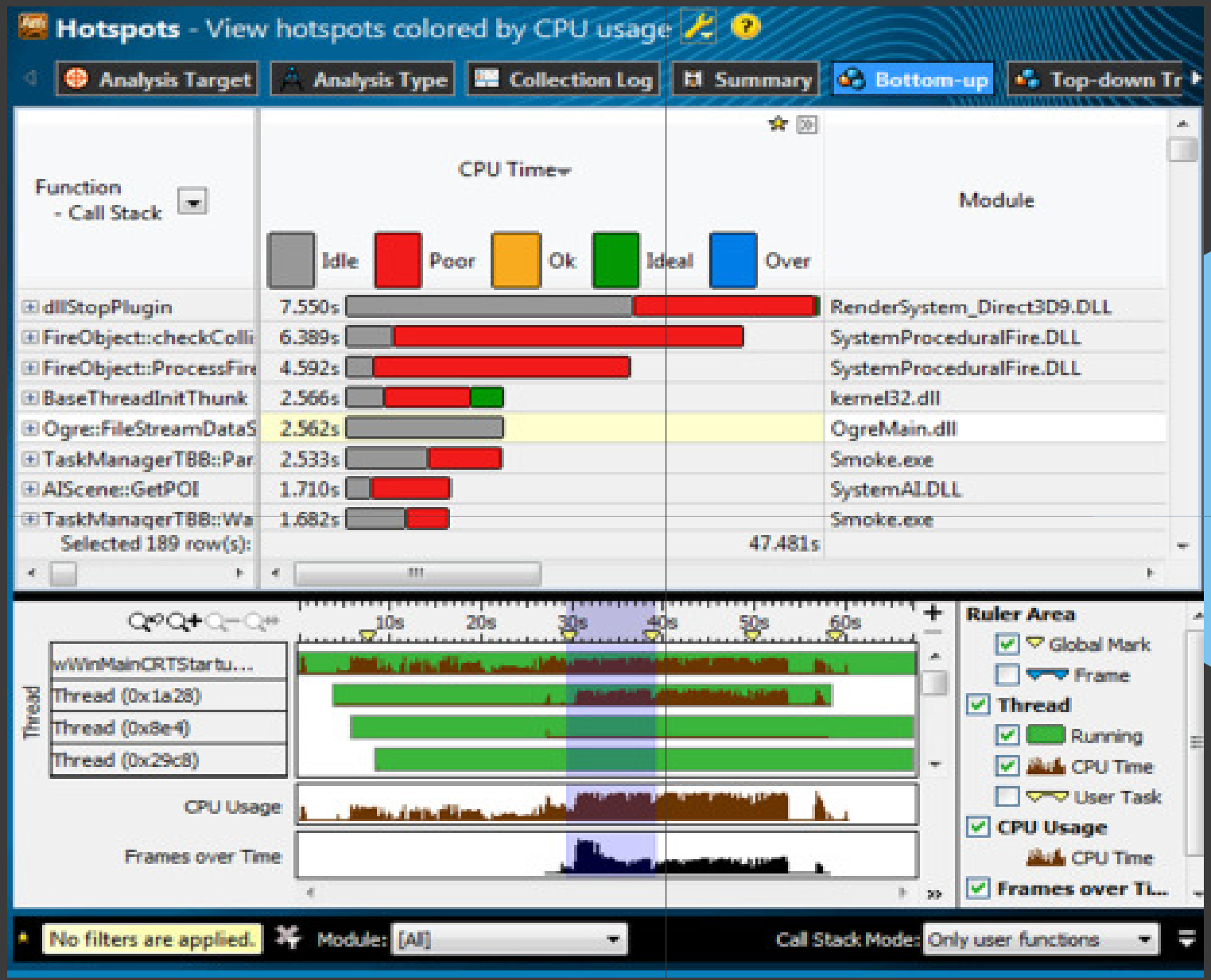
- представление результатов различных типов анализа в виде временных диаграмм;
- привязка к исходному тексту или результатам дизассемблирования.

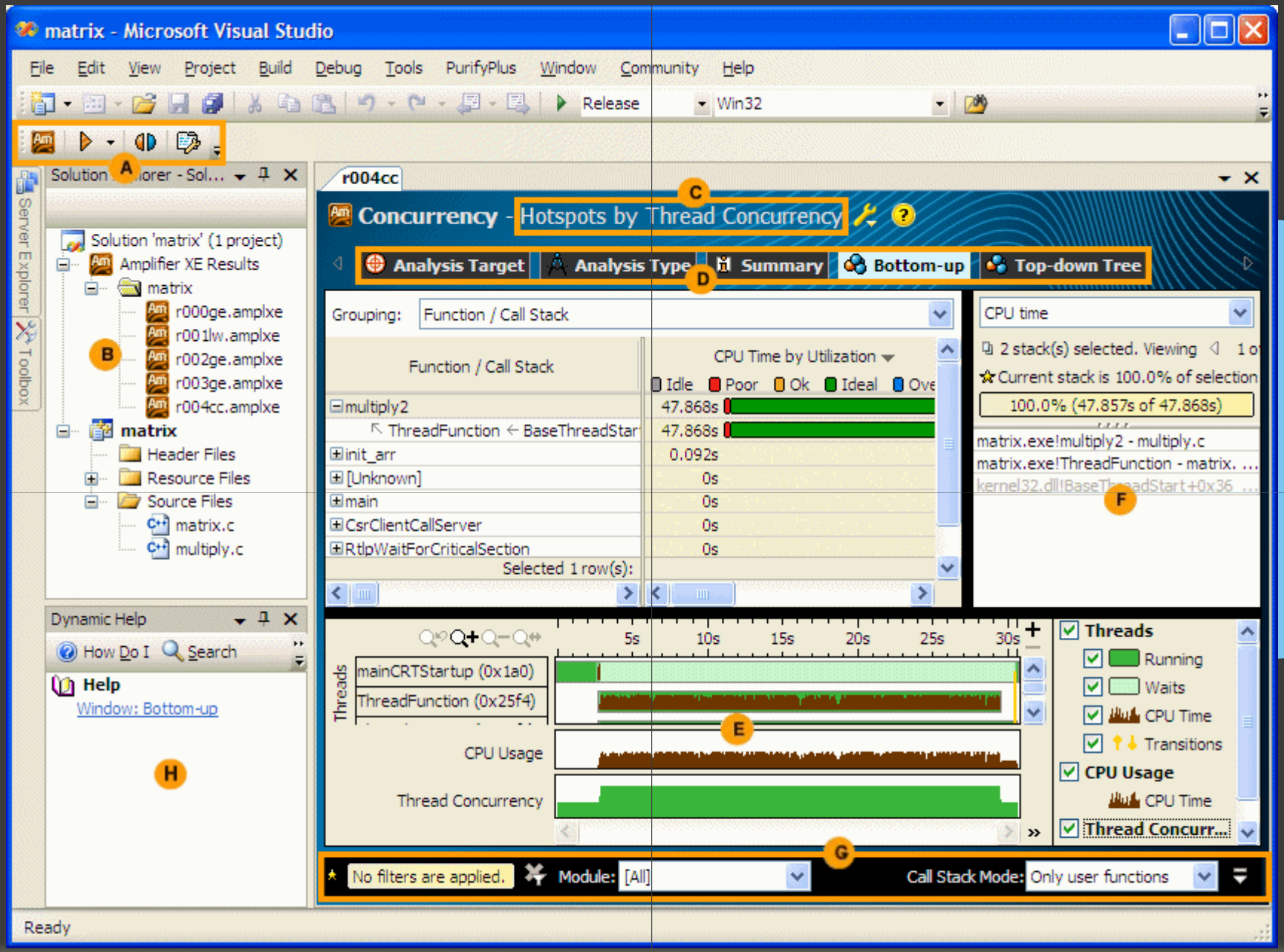
Набор заранее сгенерированных профилей анализа.

Для выполнения анализа приложения VTune потребуются:

- исполняемый (бинарный) файл, подготовленный соответствующим образом;
- исходный файл (не обязательно, но желательно, поскольку в этом случае результаты сбора статистики можно «привязать» к исходному коду программы);
- символьная информация;
- информация о номерах строк.

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.


Hotspots - View hotspots colored by CPU usage
Analysis Target
Analysis Type
Collection Log
Summary
Bottom-up
Top-down Tr
Function - Call Stack
CPU Time
Module
Idle
Poor
Ok
Ideal
Over
dllStopPlugin 7.550s RenderSystem_Direct3D9.DLL
FireObject::checkColli 6.389s SystemProceduralFire.DLL
FireObject::ProcessFire 4.592s SystemProceduralFire.DLL
BaseThreadInitThunk 2.566s kernel32.dll
Ogre::FileStreamDataS 2.562s OgreMain.dll
TaskManagerTBB::Par 2.533s Smoke.exe
AIScene::GetPOI 1.710s SystemAI.DLL
TaskManagerTBB::Wa 1.682s Smoke.exe
Selected 189 row(s): 47.481s
10s
20s
30s
40s
50s
60s
Thread
wWinMainCRTStartu...
Thread (0x1a28)
Thread (0x8e4)
Thread (0x29c8)
CPU Usage
Frames over Time
Ruler Area
Global Mark
Frame
Thread
Running
CPU Time
User Task
CPU Usage
CPU Time
Frames over Ti...
No filters are applied.
Module: [All]
Call Stack Mode: Only user functions

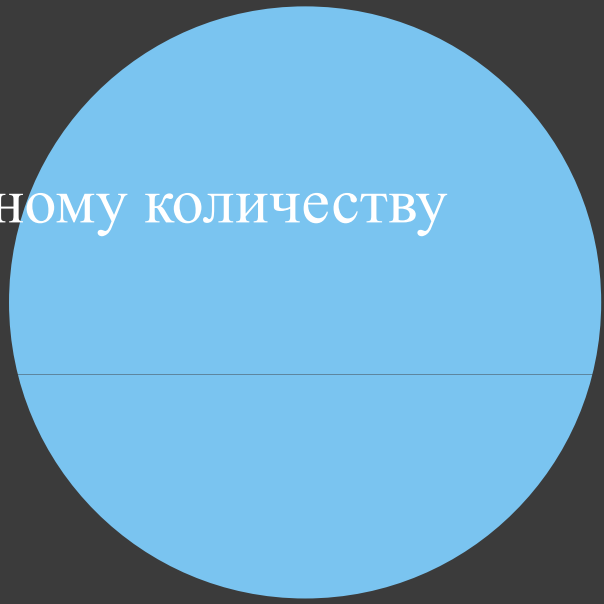

matrix - Microsoft Visual Studio
File Edit View Project Build Debug Tools PurifyPlus Window Community Help
Release
Win32
A
Solution Explorer - Sol...
Solution 'matrix' (1 project)
Amplifier XE Results
matrix
r000ge.amplxe
r001lw.amplxe
r002ge.amplxe
r003ge.amplxe
r004cc.amplxe
B
matrix
Header Files
Resource Files
Source Files
matrix.c
multiply.c
Server Explorer
Toolbox
Dynamic Help
How Do I
Search
Help
Window: Bottom-up
H
r004cc
C
Concurrency - Hotspots by Thread Concurrency
Analysis Target
Analysis Type
Summary
Bottom-up
Top-down Tree
D
Grouping: Function / Call Stack
Function / Call Stack
CPU Time by Utilization
Idle Poor Ok Ideal Ove
multiply2 47.868s
ThreadFunction ← BaseThreadStar 47.868s
init_arr 0.092s
[Unknown] 0s
main 0s
CsrClientCallServer 0s
RtlpWaitForCriticalSection 0s
Selected 1 row(s):
CPU time
2 stack(s) selected. Viewing 1 o
Current stack is 100.0% of selection
100.0% (47.857s of 47.868s)
matrix.exe!multiply2 - multiply.c
matrix.exe!ThreadFunction - matrix. ...
kernel32.dll!BaseThreadStart+0x36 ...
F
5s 10s 15s 20s 25s 30s
Threads
mainCRTStartup (0x1a0)
ThreadFunction (0x25f4)
E
CPU Usage
Thread Concurrency
Threads
Running
Waits
CPU Time
Transitions
CPU Usage
CPU Time
Thread Concurr...
G
No filters are applied.
Module: [All]
Call Stack Mode: Only user functions
Ready

Виды анализа Intel® VTune™ Amplifier XE

- ✓ Поиск функций, на выполнение которых тратится наибольшее время («хотспоты», «горячие пятна программы»);
- ✓ Поиск фрагментов программы, которые неэффективно используют процессор.
- ✓ Поиск фрагментов программы, в которых следует в первую очередь выполнить оптимизацию последовательного кода.
- ✓ Поиск фрагментов программы, в которых следует в первую очередь выполнить оптимизацию многопоточной реализации.
- ✓ Поиск объектов синхронизации, которые «плохо» влияют на производительность.
- ✓ Поиск неэффективных операций ввода-вывода.
- ✓ Поиск «узких мест», связанных с неэффективным использованием оборудования.
- ✓ Другие.

Сбор статистики:

- по времени (TBS);
- по заданному количеству событий (например, по заданному количеству ошибочно предсказанных ветвлений, EBS);
- привязка к исходному коду или дизассемблирование.



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.

«Горячие пятна»

Hotspot – это фрагмент, в котором программа проводит значительную часть времени.

Hotspots определяются в терминах событий CPU_CLK_UNHALTED.THREAD («тиков»).

2 вида анализа Intel® VTune Amplifier XE:

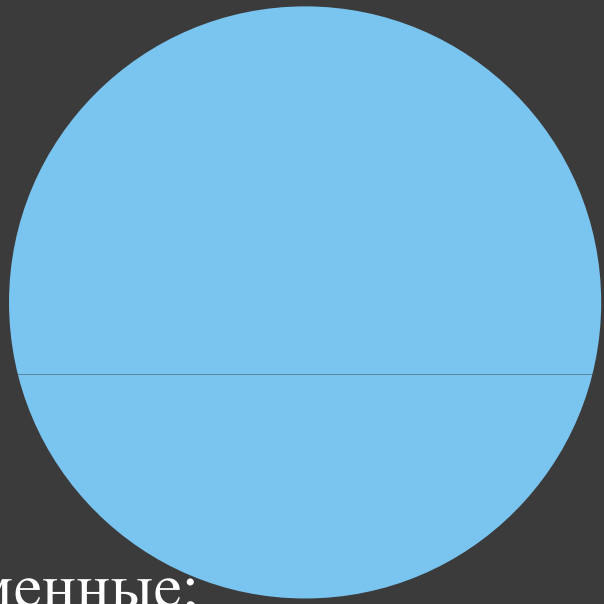
1. Hotspots
2. Lightweight hotspots

Промахи кэш-памяти

Промахи увеличивают CPI (Clocks Per Instruction).

Анализ выполняется в режиме General Exploration.

Метрики LLC_Hit, LLC_Miss.



Как уменьшить количество промахов:

- ✓ согласовать размер данных с размером кэша;
- ✓ использовать локальные по отношению к потокам переменные;
- ✓ использовать алгоритм, минимизирующий число обращений к памяти;
- ✓ другие (см. **Intel® 64 and IA-32 Architectures Optimization Reference Manual**).

Конкуренция по доступу к памяти (write sharing)

Одному ядру нужны данные, модифицированные другим ядром.
Инициируется транзакция между кэш-памятью ядра, модифицировавшего данные и ядра, которому эти данные нужны.

«Пинг-понг» увеличивает время выполнения.

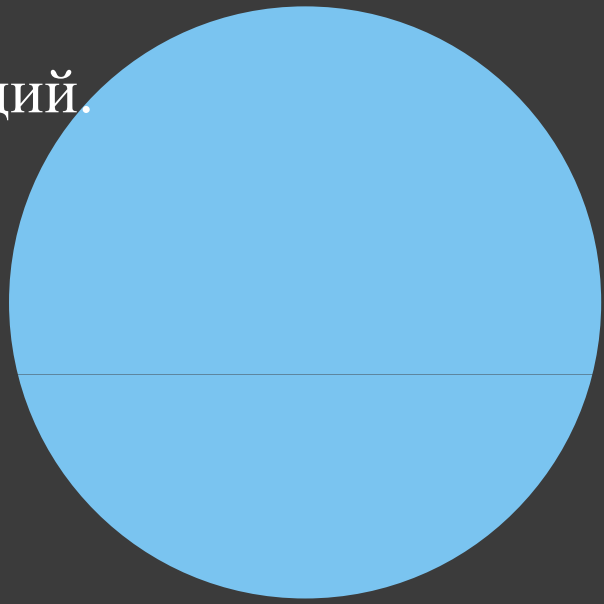
False sharing – модифицируются разные фрагменты одного набора данных, находящиеся в одной строке кэш-памяти.
True sharing – модификация одних и тех же данных.

Выравнивание данных

Отсутствие выравнивания увеличивает количество транзакций.

Профиль General Exploration.

Метрика *4K Aliasing*.

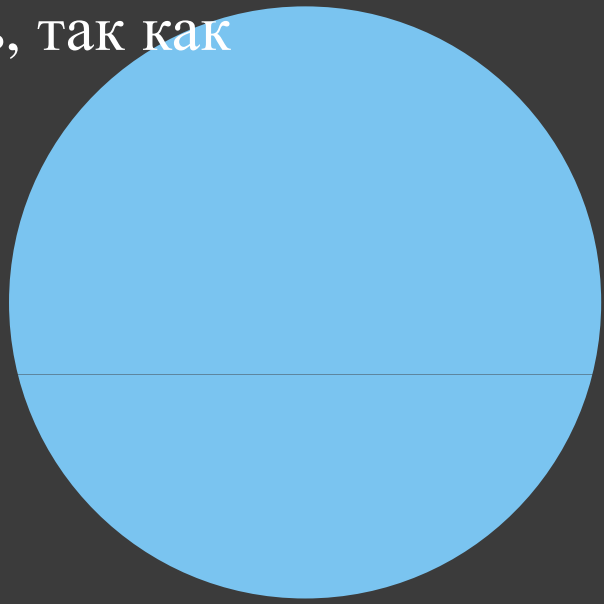


Что делать:

- ✓ использовать выравнивание данных;
- ✓ использовать «правильные» операции доступа к памяти.

Ошибки предсказания ветвлений

Ошибки предсказания ветвлений ухудшают эффективность, так как инициируются дополнительные выборки инструкций.



Профиль General Exploration.

Метрика *Branch Mispredict*.

Ошибки предсказания ветвлений есть всегда!

Что делать, чтобы минимизировать число ошибок предсказания ветвлений:
✓ использовать при компиляции оптимизацию на основе профилирования.

Виды анализа Intel® Vtune Amplifier XE

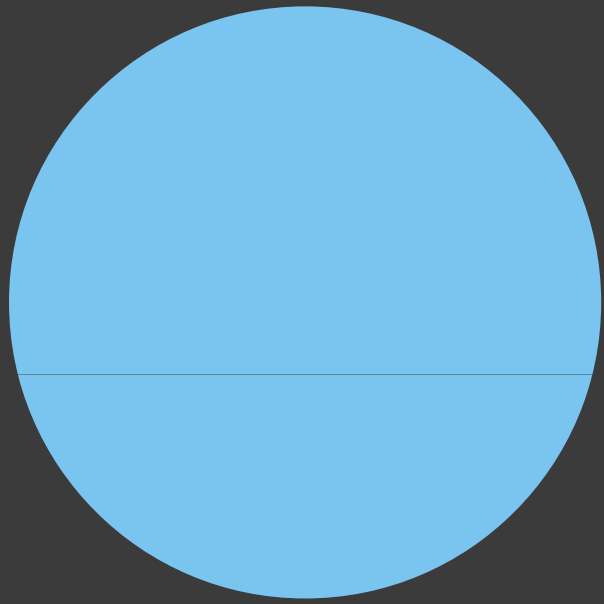
Algorithm analysis

- **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.

Виды анализа Intel® Vtune Amplifier XE

Intel Core 2 Processor Analysis

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

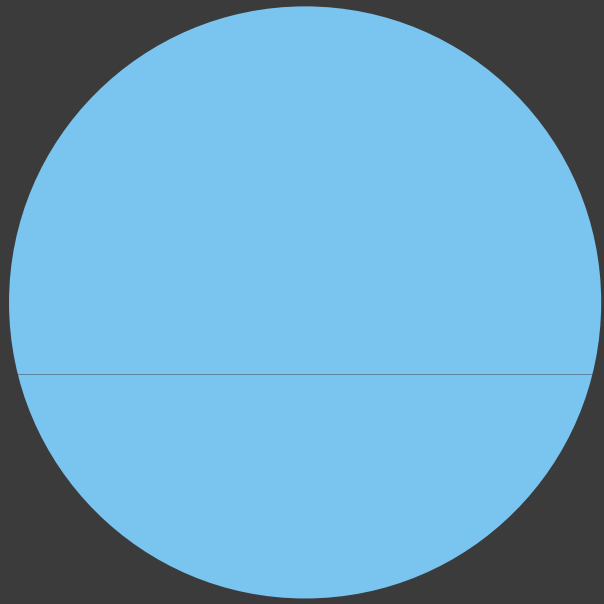


Виды анализа Intel® Vtune Amplifier XE

Nehalem / Westmere Analysis

(Core i5, Core i7, Xeon E7, …)

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



Виды анализа Intel® Vtune Amplifier XE

Sandy Bridge / Ivy Bridge / Haswell Analysis

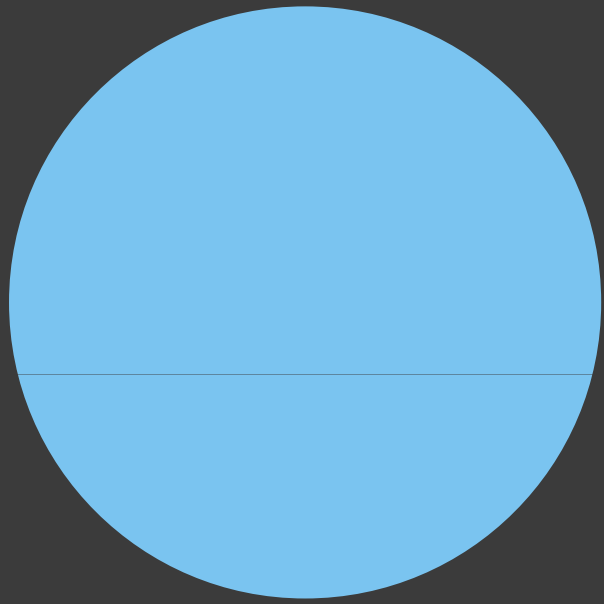
(Intel® Xeon™ E5, Core i3, Core i5, Core i7, …)

- **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**.

Виды анализа Intel® Vtune Amplifier XE

Intel Atom Processor Analysis

General Exploration.

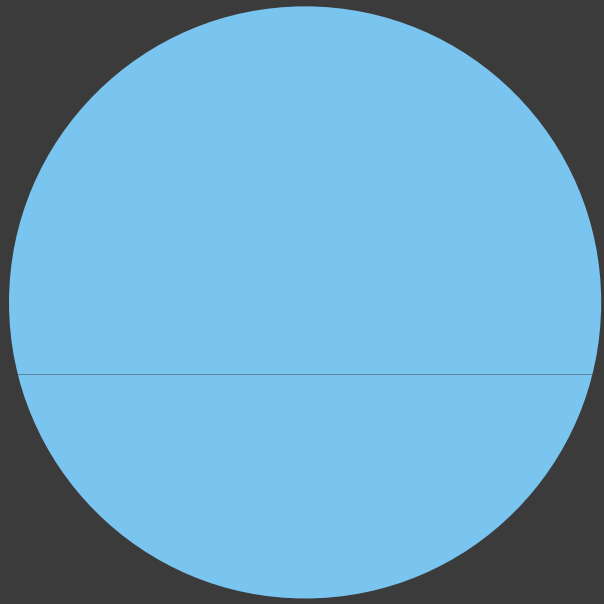


Виды анализа Intel® Vtune Amplifier XE

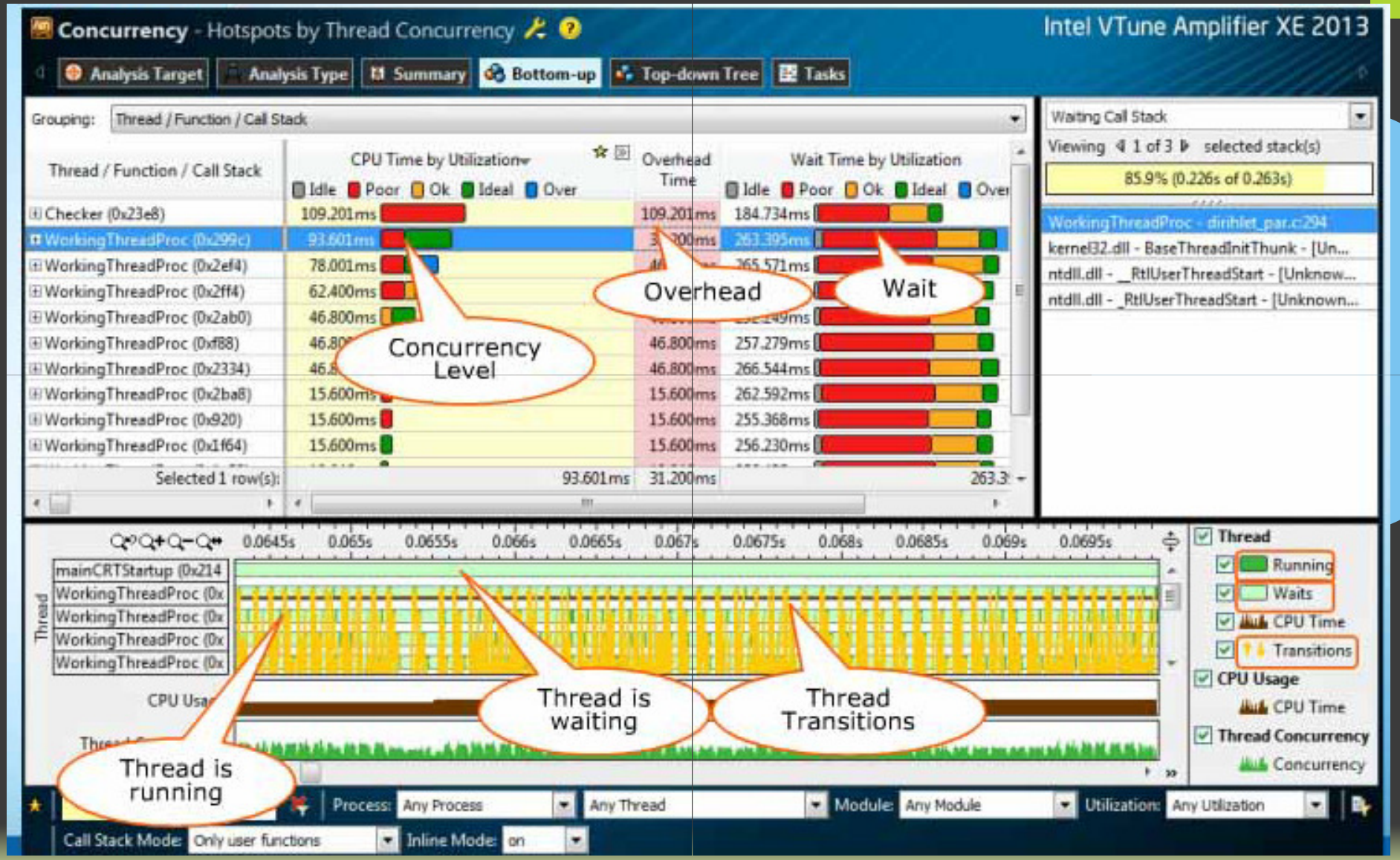
Knights Corner Platform Analysis

(Intel® Xeon™ Phi)

- Hotspots.
- General Exploration.
- Bandwidth.

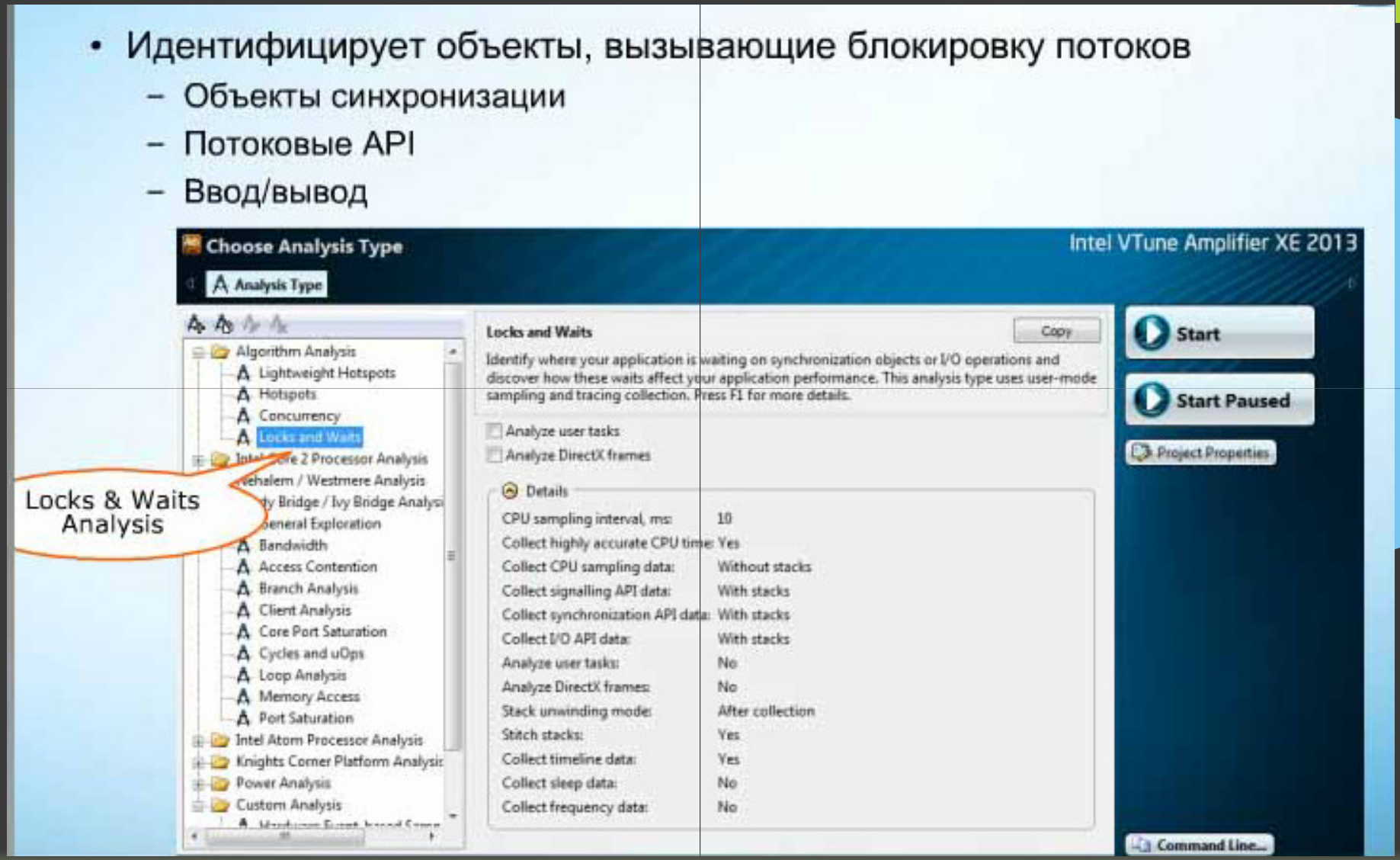


Анализ эффективности многопоточного параллелизма

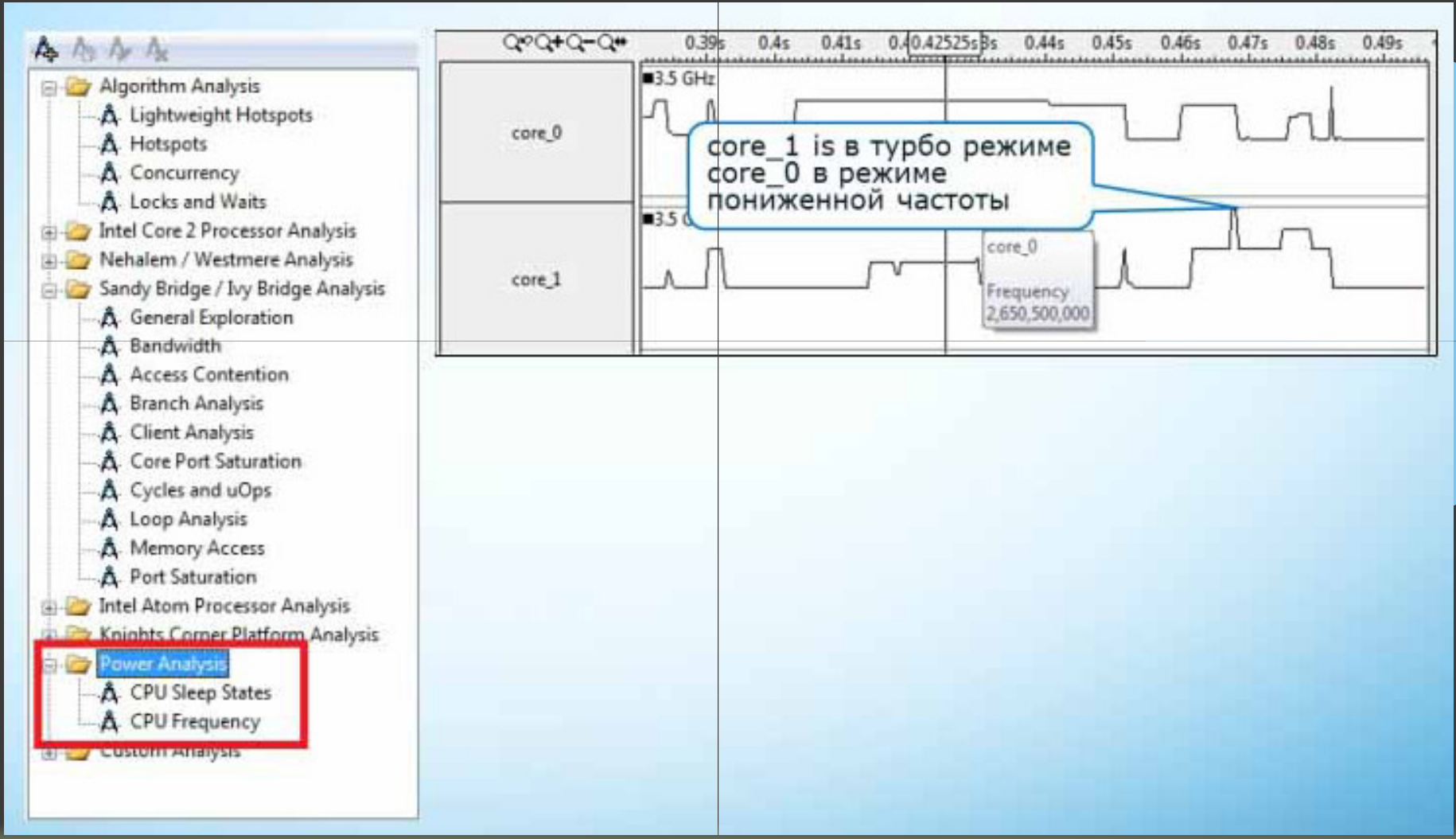


Анализ блокировок и простоев

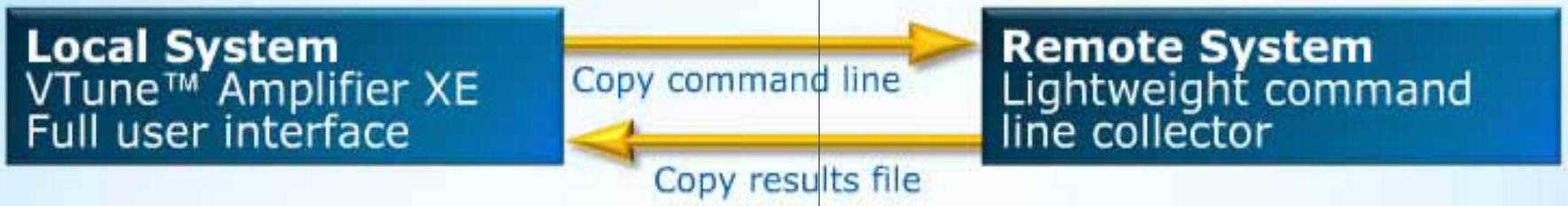
- Идентифицирует объекты, вызывающие блокировку потоков
  - Объекты синхронизации
  - Потоковые API
  - Ввод/вывод



Анализ энергопотребления
(CPU Sleep State, CPU frequency)



Удалённый анализ



1. Конфигурируйте анализ локально в GUI
2. Копируйте командную строку
3. Вставьте команду в консоль удалённой машины
4. Скопируйте файлы результатов на локальную машину
5. Откройте файл в локальном GUI

- Минимальное влияние анализа на производительность
- Простая конфигурация из GUI
- Простой анализ результатов