



NVIDIA Parallel Nsight (aka Nexus)

NVIDIA DevTech | Anton Obukhov



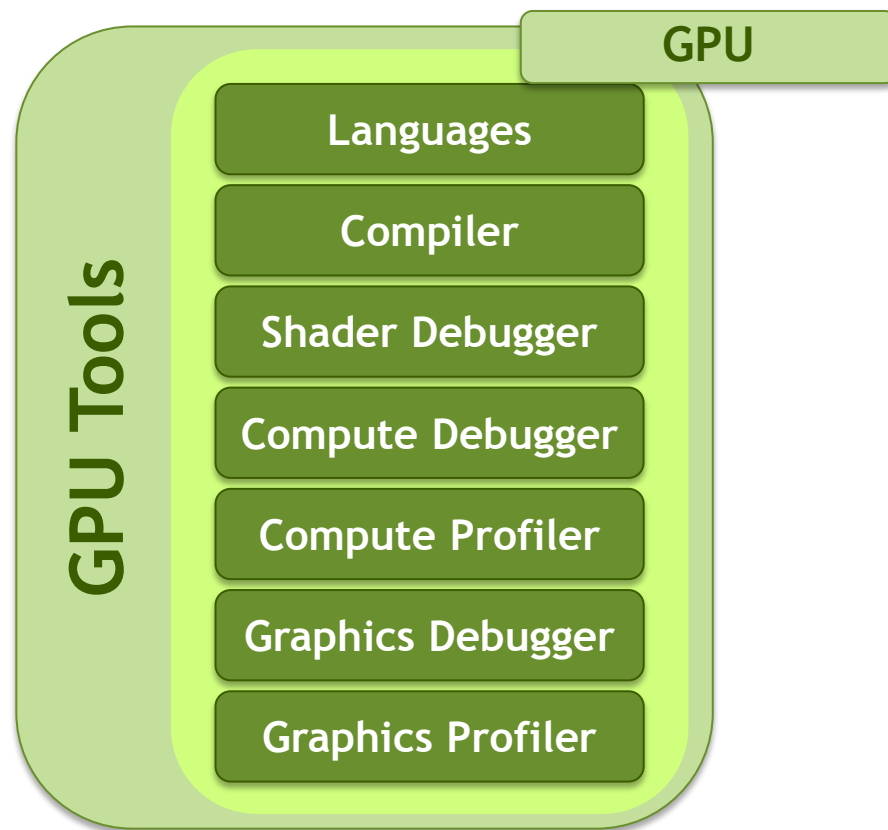
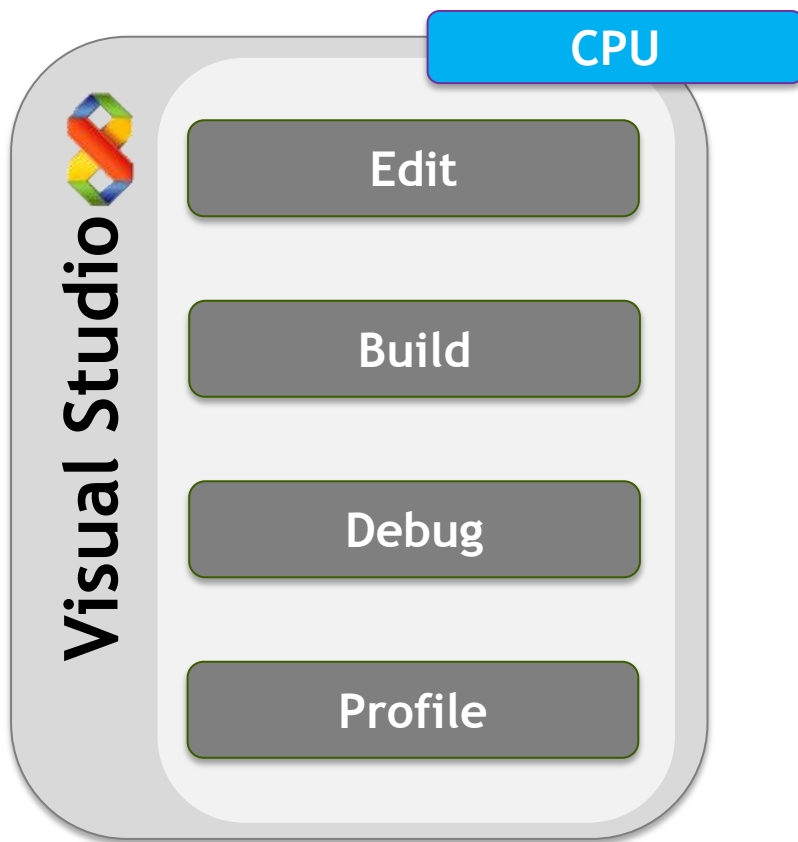
NVIDIA.

Что такое Nexus?

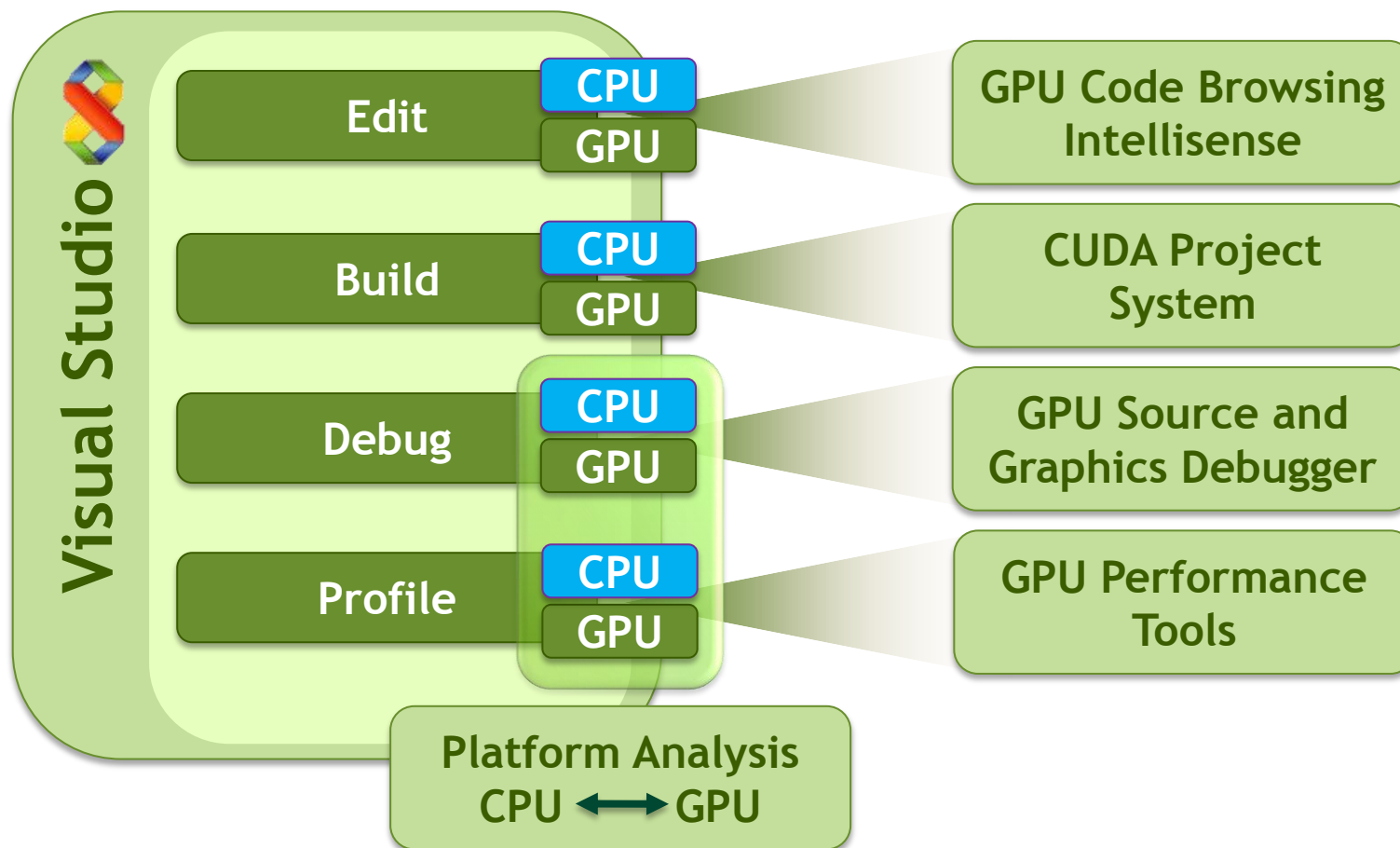
NVIDIA Nexus - Среда разработки приложений для гетерогенных платформ

Nexus интегрируется в Microsoft Visual Studio с целью обеспечить **гетерогенность** разработки

Windows Development Environment



... чего хотят разработчики



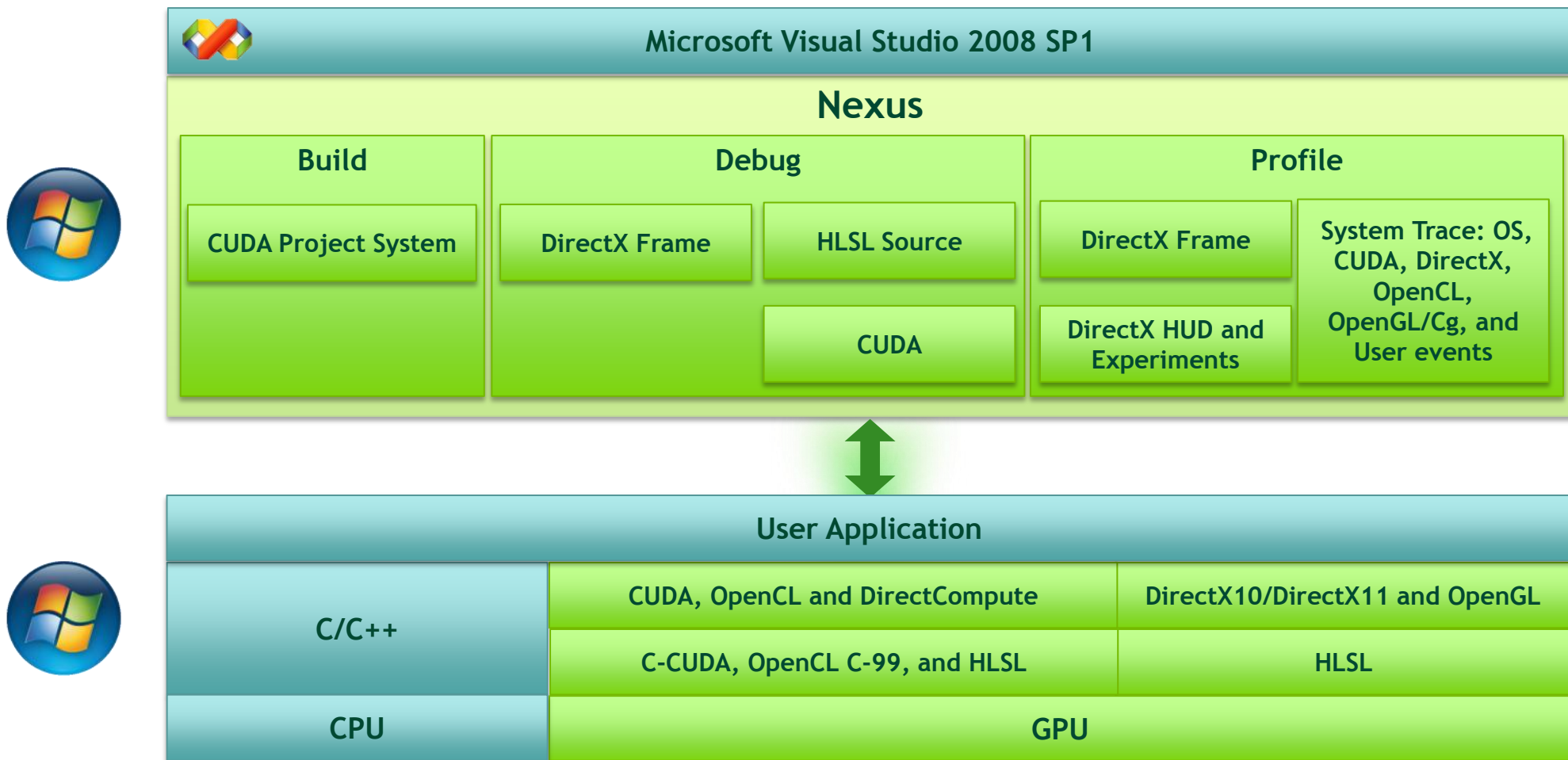
NVIDIA Nexus

Возможность разработки для CPU и GPU в едином цикле разработки приложения

- Отладка кода на самом GPU
- Анализ активности
- Интеграция с Visual Studio 2008



Nexus Overview



Native GPU Debugging

- Отладка на GPU, без эмуляции
- Точное соответствие результатов FPU
- GPU останавливается на breakpoint
- Исследование различных пространств памяти
- Высокая эффективность/скорость отладки

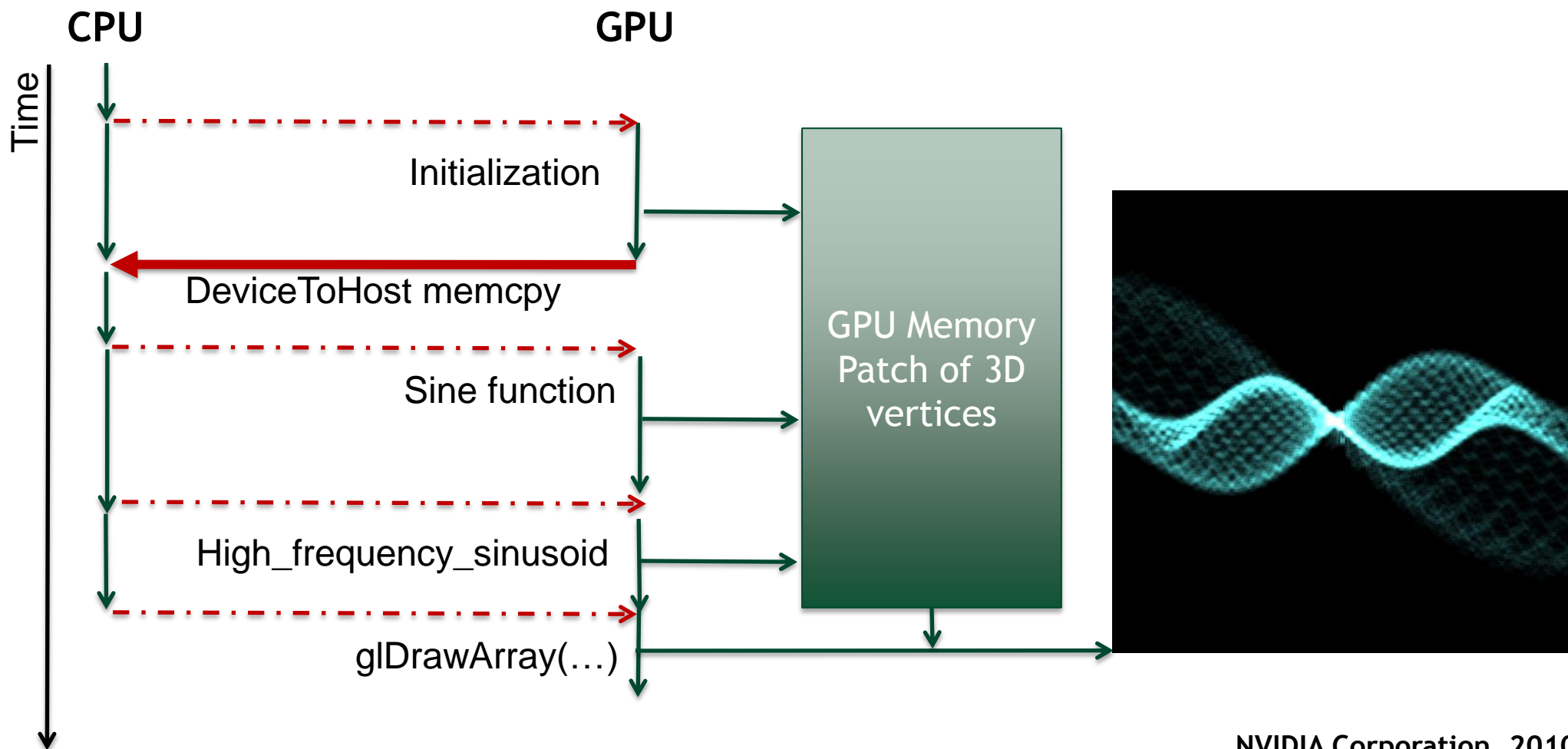
Nexus CUDA Debugger

- Сериализация запусков kernel'ов
- Контроль на уровне Warp'ов и пошаговое исполнение
- Visual Studio debugger фокусирование на треде
- Поддержка встроенных окон Visual Studio
 - Variables, Watch, Memory, Registers, Modules, Processes, and Threads

Nexus CUDA Debugger

- Улучшения Nexus Parallel Debugger
 - Несколько точек останова по данным и на инструкциях
 - Вычисление выражений «на лету»
 - CUDA focus picker
 - Вычисление выражений поддерживает иерархию памяти
- CUDA memory (bounds) checker

Heterogeneous Computing Example



Nexus C-CUDA Debugger Demo

The screenshot displays the NVIDIA Nexus C-CUDA Debugger interface within Microsoft Visual Studio. The main window shows the memory dump for address 0x00110000, displaying a grid of floating-point values. A disassembly window is open, showing the code for `simpleGL_kernel.cu`. A dialog box titled "NVIDIA Nexus - CUDA Focus Picker" is overlaid, showing dimensions for a block (0, 0, 0) and thread (0, 0, 0). The dialog also includes examples for block and thread indices. The bottom of the interface shows the Locals window with variables like `blockDim`, `gridDim`, `x`, `y`, `idx`, `g_data`, `width`, and `factor`. The Breakpoints window shows several breakpoints set for the `simpleGL_kernel.cu` file.

Process: [3772] GPU - simpleGL.exe Thread: [75972448] <No Name> Stack Frame: Module: 112636592 - [0] _Z1

Memory1

Address: 0x00110000

Address	Value	Value	Value	Value	Value	Value	Value	Value	Value
0x00110040	-1.9375000	0.050880503	-2.0000000	1.0000000	-1.9218750	0.051616980	-2.0000000
0x0011005C	1.0000000	-1.9062500	0.050274149	-2.0000000	1.0000000	-1.8906250	0.047362056
0x00110078	-2.0000000	1.0000000	-1.8750000	0.043980788	-2.0000000	1.0000000	-1.8593750
0x00110094	0.041408714	-2.0000000	1.0000000	-1.8437500	0.040618271	-2.0000000	1.0000000
0x001100B0	-1.8281250	0.041908298	-2.0000000	1.0000000	-1.8125000	0.044731073	-2.0000000
0x001100CC	1.0000000	-1.7968750	0.048176695	1.0000000	-1.7812500	0.044731073	-2.0000000
0x001100E8	-2.0000000	1.0000000	-1.7656250	0.044731073	-2.0000000	1.0000000	-1.7500000

Disassembly: simpleGL_kernel.cu

```
/** @param data data in global memory
 *  @param width width of data in global memory
 */
__global__ void init_array(float4 *g_data, unsigned int width)
{
    unsigned int x = blockIdx.x*blockDim.x + threadIdx.x;
    unsigned int y = blockIdx.y*blockDim.y + threadIdx.y;

    int idx = y*width+x;

    for(int i=0;i<num_iterations;i++)
    {
        g_data[idx].x += (*factor).x; // non-coalesced on purpose, to burn
        g_data[idx].y += (*factor).y; // non-coalesced on purpose, to burn
        g_data[idx].z += (*factor).z; // non-coalesced on purpose, to burn
        g_data[idx].w += (*factor).w; // non-coalesced on purpose, to burn
    }
}
```

NVIDIA Nexus - CUDA Focus Picker

Block: 0, 0, 0 Dimensions: 1, 64, 1

Thread: 0, 0, 0 Dimensions: 256, 1, 1

Examples

- #129 for block index 129
- 10 for coordinates 10, 0
- 10, 5 for coordinates 10, 5

OK Cancel

Summary

Device 0

12636592 c:/ProgramData/NVIDIA Corporation/NVIDIA GPU Comput...

0, 0, 0 ! Warp Mask: 0x000000FF

0 [0, 0, 0] ! Active Mask: 0xFFFFFFFF, PC: 0x000D0098, simplegl_ker

1 [32, 0, 0] ! Active Mask: 0xFFFFFFFF, PC: 0x000D0098, simplegl_ker

Warp 2 [64, 0, 0] ! Active Mask: 0xFFFFFFFF, PC: 0x000D0098, simplegl_ker

Warp 3 [96, 0, 0] ! Active Mask: 0xFFFFFFFF, PC: 0x000D0098, simplegl_ker

Warp 4 [128, 0, 0] ! Active Mask: 0xFFFFFFFF, PC: 0x000D0098, simplegl_ker

Warp 5 [160, 0, 0] ! Active Mask: 0xFFFFFFFF, PC: 0x000D0098, simplegl_ker

Warp 6 [192, 0, 0] ! Active Mask: 0xFFFFFFFF, PC: 0x000D0098, simplegl_ker

Warp 7 [224, 0, 0] ! Active Mask: 0xFFFFFFFF, PC: 0x000D0098, simplegl_ker

Block 1 [0, 1, 0] ! Warp Mask: 0x000000FF

Locals

Name	Value	Type
blockDim	{x = 256, y = 1, z = 1}	const dim
gridDim	{x = 1, y = 64, z = 1}	const dim
x	0	unsigned
y	???	unsigned
idx	???	int
g_data	0x00110000 {x = -2, y = -0.2528252, z = -2, w = 1}	_device_
width	256	_shared_
factor	0x00210000 {x = 0, y = 0, z = 0, w = 0}	_device_

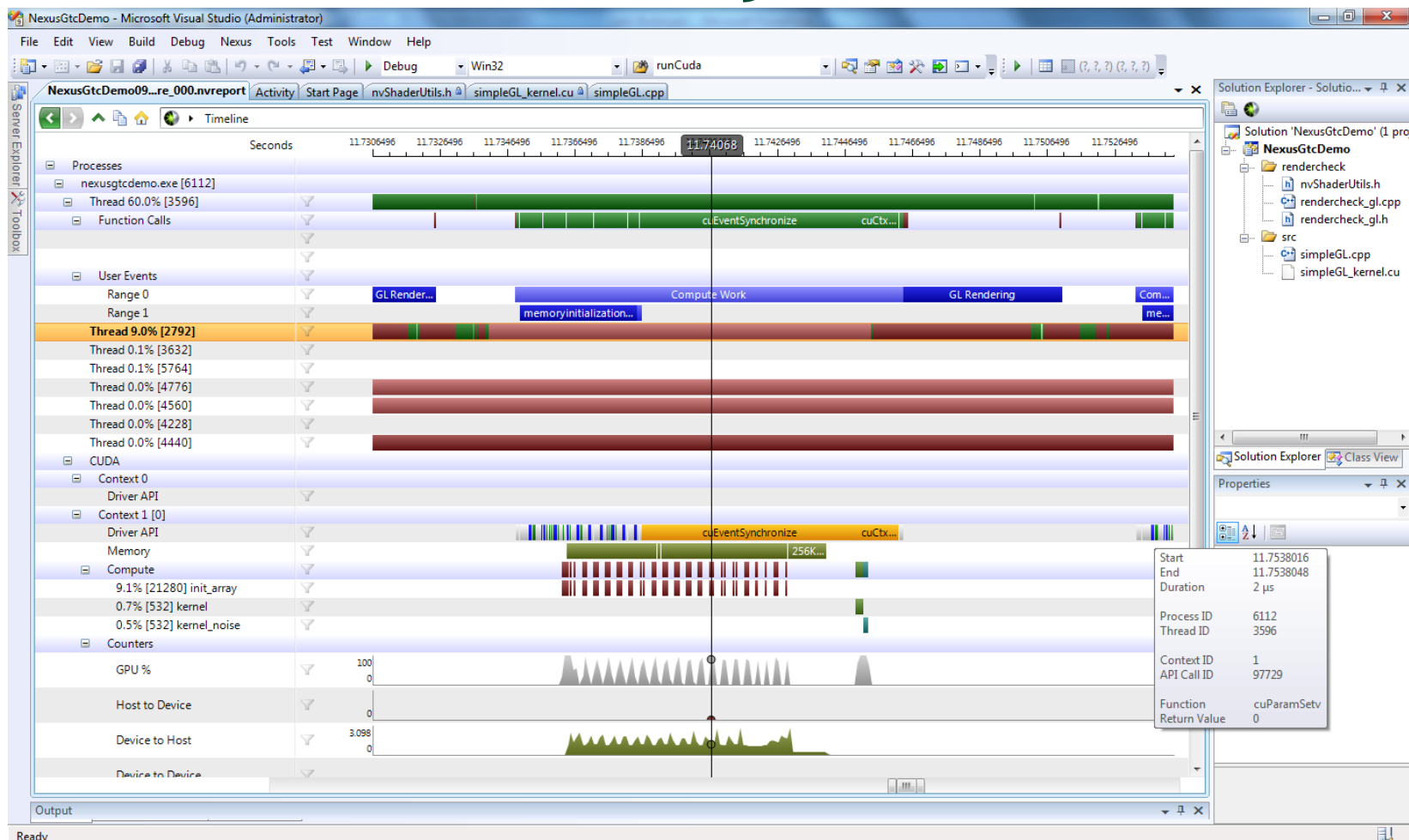
Breakpoints

Name	Condition	Hit Count
When '0x0011008c' changes (4 bytes)	(no condition)	break always (currently 0)
simpleGL_kernel.cu, line 119	(no condition)	break always
simpleGL_kernel.cu, line 29	(no condition)	break always (currently 0)

Platform Analysis

- Сбор данных об активности
 - OS - process, thread and module events
 - CPU API Trace - CUDA driver API, DirectX, OpenGL, OpenCL, Cg2.2 and User Events
 - GPU Task Trace - C-CUDA and OpenCL launches and memory copies
- Отчетность о времени, событиях, вызовах, API, загрузке GPU

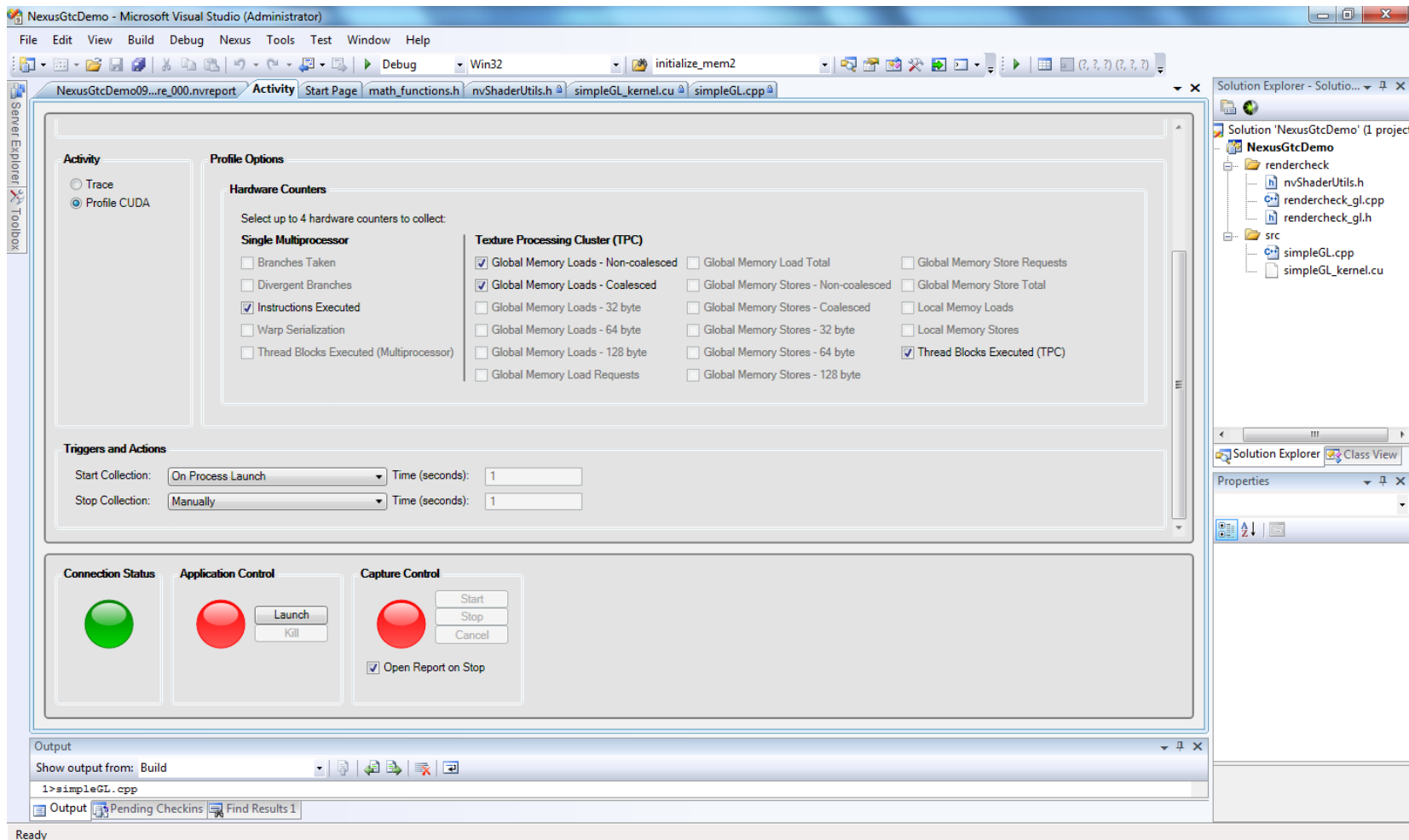
Nexus Platform Analysis Demo



Профилирование CUDA

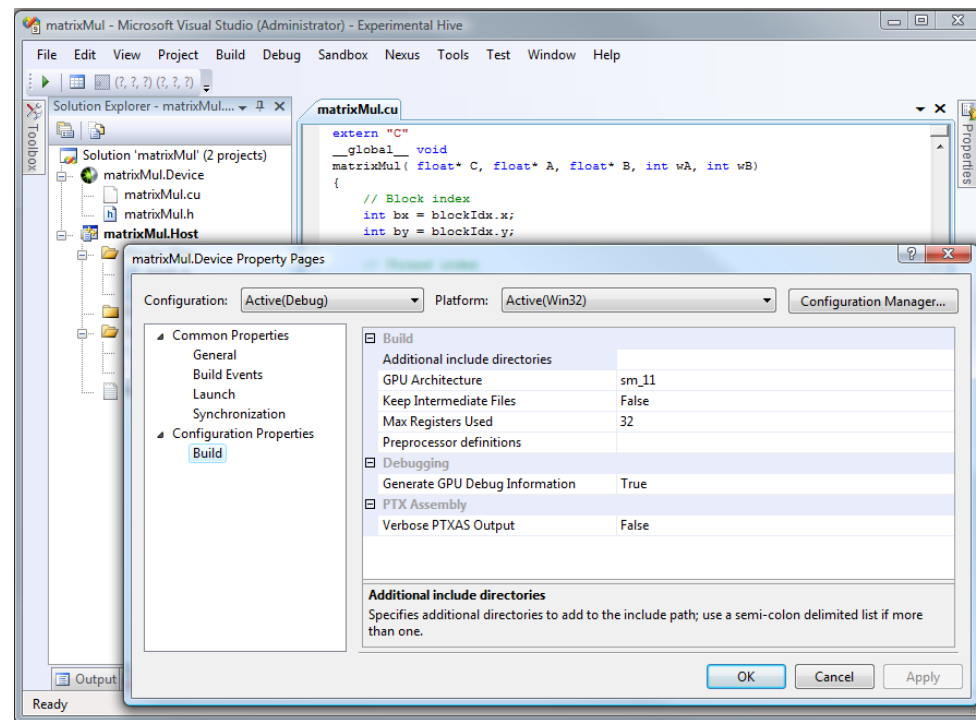
- Базовая информация
 - Memory coalescing
 - Branch divergence
 - Instructions executed
 - Warp serialization
- Используются счетчики инструкций и часы на GPU

Nexus CUDA Profiler



Nexus Build

- C++ project system
 - Nexus options
 - CUDA vsprop files
- CUDA project system
 - NVCC build integration
 - Error reporting
 - Debugger session



Поддерживаемые OS



Windows Vista SP1

32 bit
64 bit
32 on 64

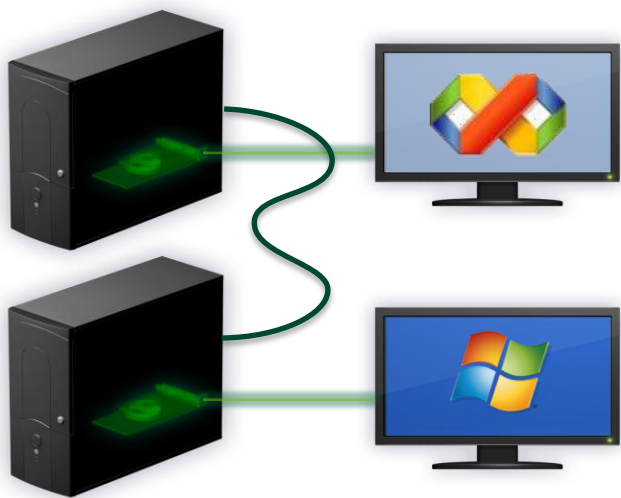


Windows 7

Системные требования

Удаленная отладка

Desktop



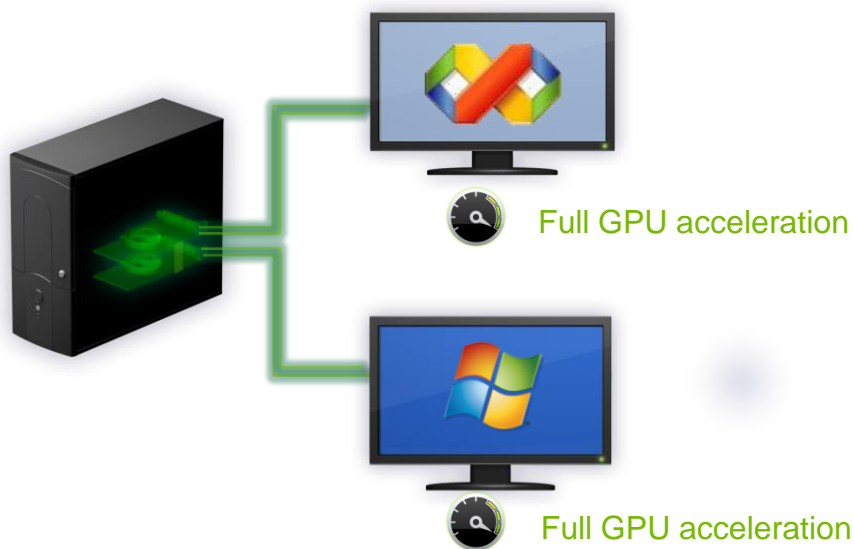
Mobile



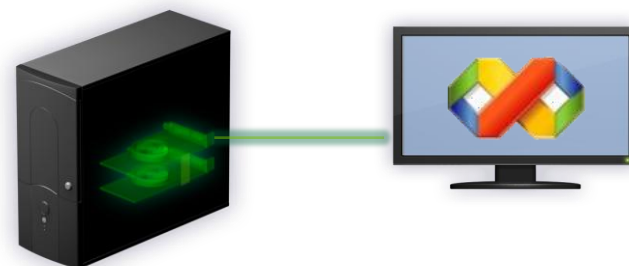
Системные требования

Локальная отладка

SLI Multi-OS



Multi-GPU



Живое Демо & Q/A



- Настройка Nexus Project в Visual Studio
- Отладка с Nexus
- Анализ и профилирование с Nexus
- Nexus Memory Checker

Становитесь зарегистрированным разработчиком уже сегодня:

<http://developer.nvidia.com/object/nexus.html>