Visual Studio IDE for C++ Developers – What's New

Sumit Kumar
Program Manager
Visual C++ Team
Microsoft Corporation

Goal

Help YOU, the C++ developers, become more productive with developing C++ code using Visual Studio IDE

Outline

- Productivity Features in the Editor
- Productivity Features in the Overall IDE
- Code Analysis
- Debugging
- Team oriented features

Demo

Demo Summary

Productivity in the Editor

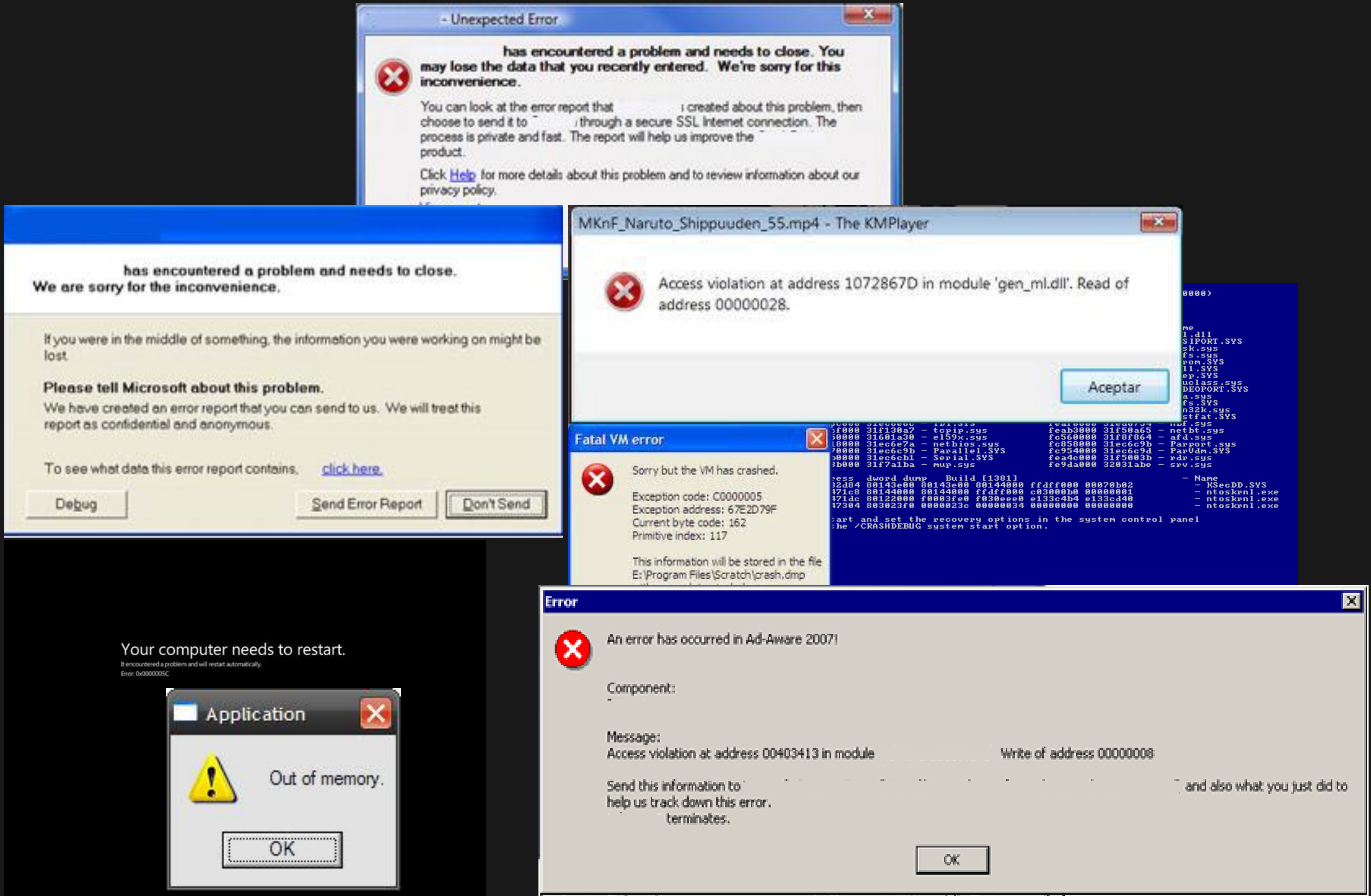
- Enhanced IntelliSense
- Semantic Colorization
- Reference Highlighting
- Code Snippets
- Find
- XML Doc Comments
- Diffing

Demo Summary

Productivity in the Overall IDE

- Simplified UI
- New Solution Explorer
- Dependency Graphs
- Improved Document Management
- Preview Tabs
- Search Everywhere

Code Analysis



- Unexpected Error

has encountered a problem and needs to close. You may lose the data that you recently entered. We're sorry for this inconvenience.

You can look at the error report that created about this problem, then choose to send it to through a secure SSL Internet connection. The process is private and fast. The report will help us improve the product.

Click Help for more details about this problem and to review information about our privacy policy.

has encountered a problem and needs to close. We are sorry for the inconvenience.

If you were in the middle of something, the information you were working on might be lost.

Please tell Microsoft about this problem.

We have created an error report that you can send to us. We will treat this report as confidential and anonymous.

To see what data this error report contains, click here.

MKnF_Naruto_Shippuuden_55.mp4 - The KMPlayer

Access violation at address 1072867D in module 'gen_ml.dll'. Read of address 00000028.

```
0000)

me
l.dll
SIPORT.SYS
sk.sys
fs.sys
rom.SYS
ll.SYS
ep.SYS
uclass.sys
DEOPORT.SYS
a.sys
fs.SYS
n32k.sys
stfat.SYS
                                                   ndf.sys
cf000 31f130a7 - tcpip.sys          feab3000 31f50a65 - netbt.sys
50000 31601a30 - e159x.sys          fc560000 31f8f864 - afd.sys
18000 31ec6e7a - netbios.sys        fc858000 31ec6c9b - Parport.sys
70000 31ec6c9b - Parallel.SYS       fc954000 31ec6c9d - ParVdm.SYS
b0000 31ec6cb1 - Serial.SYS         fea4c000 31f5003b - rdr.sys
3b000 31f7a1ba - mup.sys            fe9da000 32031abe - srv.sys

ess  dword dump   Build [1381]                          - Name
32d84 80143e00 80143e00 80144000 ffdff000 00070b02        - KSecDD.SYS
471c8 80144000 80144000 ffdff000 c03000b0 00000001        - ntoskrnl.exe
471dc 80122000 f0003fe0 f030eee0 e133c4b4 e133cd40        - ntoskrnl.exe
47304 803023f0 0000023c 00000034 00000000 00000000        - ntoskrnl.exe

tart and set the recovery options in the system control panel
the /CRASHDEBUG system start option.
```

Fatal VM error

Sorry but the VM has crashed.

Exception code: C0000005
Exception address: 67E2D79F
Current byte code: 162
Primitive index: 117

This information will be stored in the file
E:\Program Files\Scratch\crash.dmp

Your computer needs to restart.

It encountered a problem and will restart automatically.

Error: 0x0000005C

Application

Out of memory.

Error

An error has occurred in Ad-Aware 2007!

Component:
-

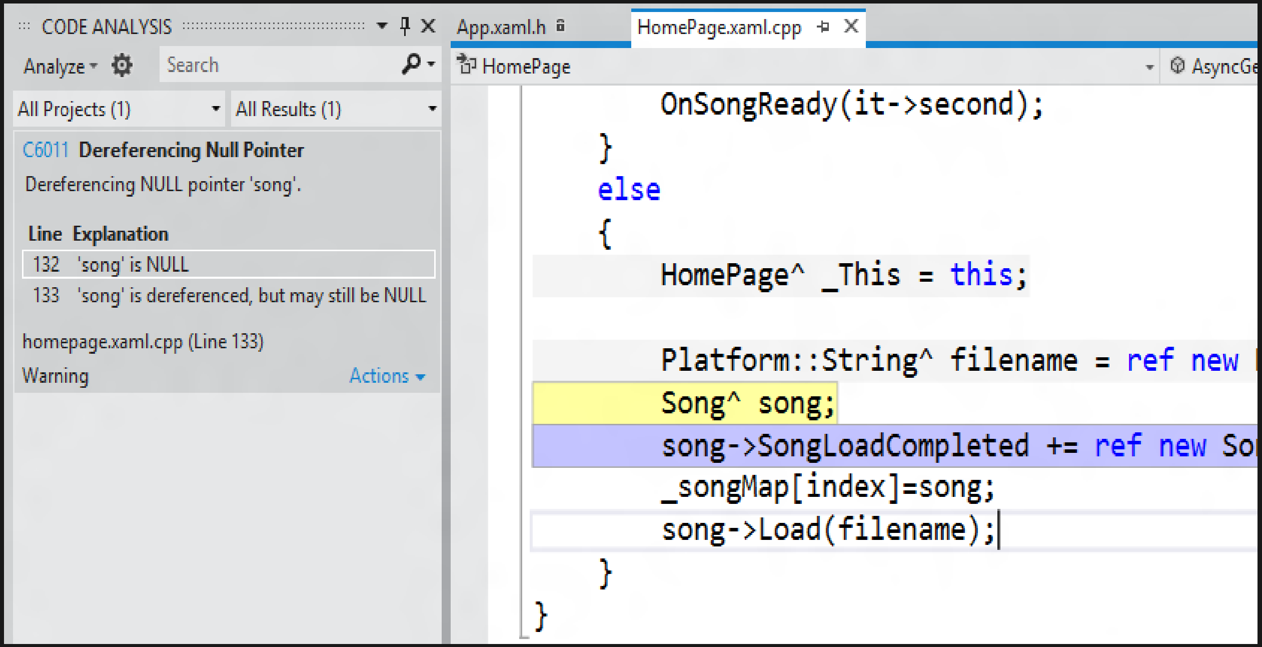
Message:
Access violation at address 00403413 in module Write of address 00000008

Send this information to and also what you just did to help us track down this error.
terminates.

Demo

Code analysis

- Improved accuracy and breadth of coverage
- Key events to help diagnose problems easier
- New code analysis window for easy management of results
- Available in all VS SKUs (including Express)



C6011 **Dereferencing Null Pointer**

Dereferencing NULL pointer 'song'.

Line	Explanation
132	'song' is NULL
133	'song' is dereferenced, but may still be NULL

homepage.xaml.cpp (Line 133)

Warning

```
            OnSongReady(it->second);
        }
        else
        {
            HomePage^ _This = this;

            Platform::String^ filename = ref new 
            Song^ song;
            song->SongLoadCompleted += ref new So
            _songMap[index]=song;
            song->Load(filename);
        }
}
```

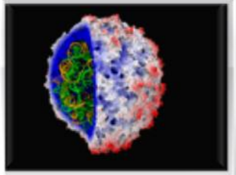
Parallel and GPU Debugging

- C++ AMP Primer
- GPU Debugging features

The Power of Heterogeneous Computing

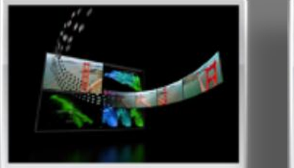
146X

Interactive visualization of volumetric white matter connectivity



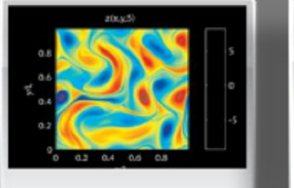
36X

Ionic placement for molecular dynamics simulation on GPU



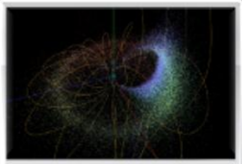
19X

Transcoding HD video stream to H.264



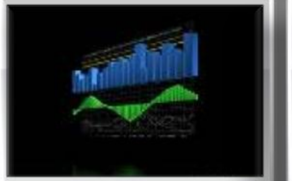
17X

Simulation in Matlab using .mex file CUDA function



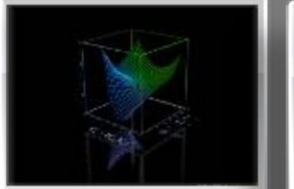
100X

Astrophysics N-body simulation



149X

Financial simulation of LIBOR model with swaptions



47X

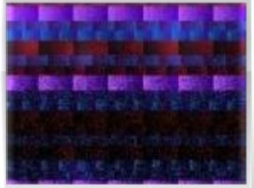
GLAME@lab: An M-script API for linear Algebra operations on GPU

20X

Ultrasound medical imaging for cancer diagnostics

24X

Highly optimized object oriented molecular dynamics

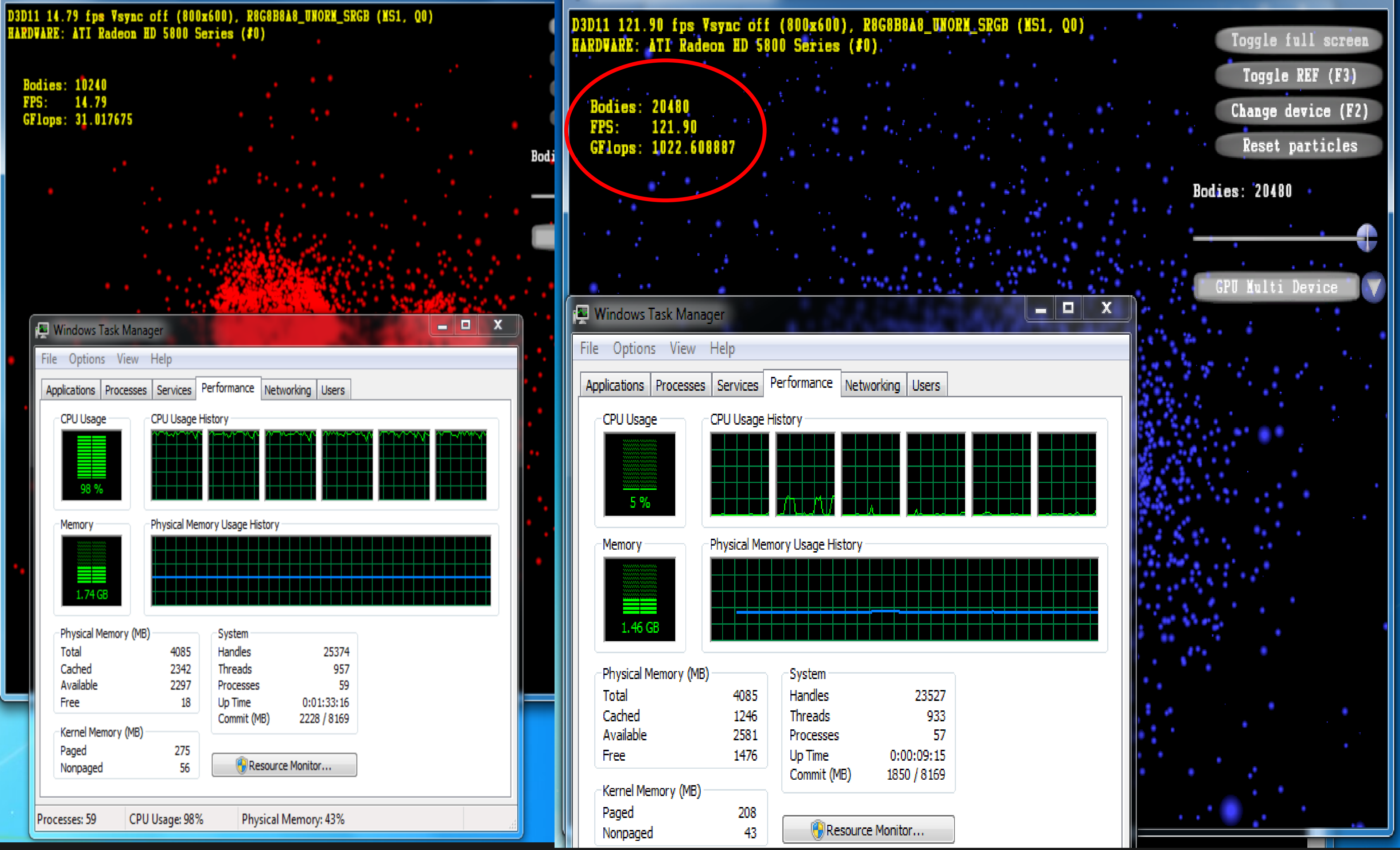


30X

Cmatch exact string matching to find similar proteins and gene sequences

source

NVIDIA



```
D3D11 14.79 fps Vsync off (800x600), R8G8B8A8_UNORM_SRGB (MS1, Q0)
HARDWARE: ATI Radeon HD 5800 Series (#0)

Bodies: 10240
FPS:    14.79
GFlops: 31.017675
```

CPU Usage: 98 %
Memory: 1.74 GB

Physical Memory (MB)	
Total	4085
Cached	2342
Available	2297
Free	18

Kernel Memory (MB)	
Paged	275
Nonpaged	56

System	
Handles	25374
Threads	957
Processes	59
Up Time	0:01:33:16
Commit (MB)	2228 / 8169

Processes: 59 | CPU Usage: 98% | Physical Memory: 43%

```
D3D11 121.90 fps Vsync off (800x600), R8G8B8A8_UNORM_SRGB (MS1, Q0)
HARDWARE: ATI Radeon HD 5800 Series (#0)

Bodies: 20480
FPS:    121.90
GFlops: 1022.608887
```

Bodies: 20480
GPU Multi Device

CPU Usage: 5 %
Memory: 1.46 GB

Physical Memory (MB)	
Total	4085
Cached	1246
Available	2581
Free	1476

Kernel Memory (MB)	
Paged	208
Nonpaged	43

System	
Handles	23527
Threads	933
Processes	57
Up Time	0:00:09:15
Commit (MB)	1850 / 8169

N-Body Simulation

C++ AMP

- Part of Visual C++
- Visual Studio integration
- STL-like library for multidimensional data
- Builds on Direct3D
- An open specification

performance

productivity

portability

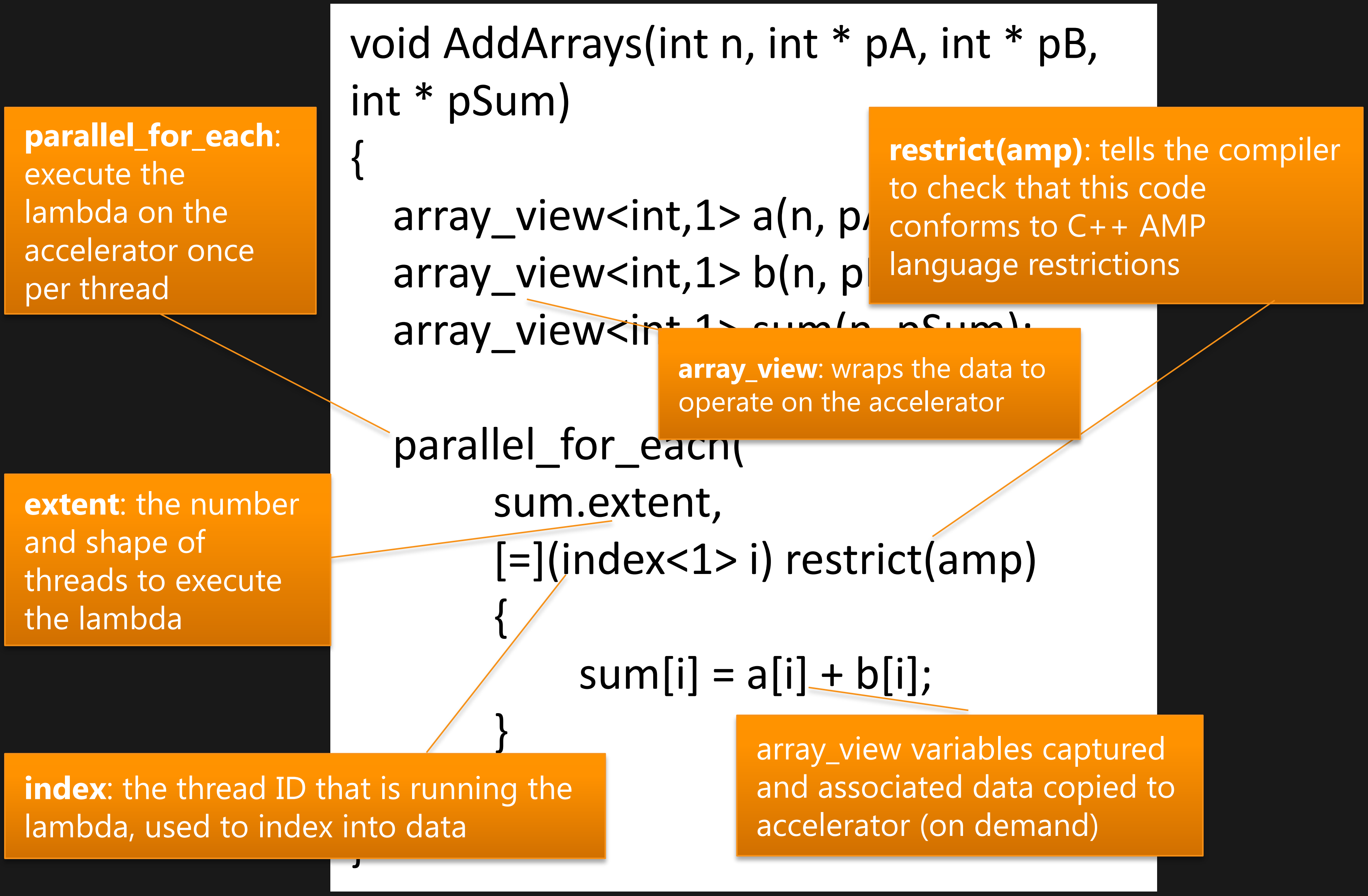
Microsoft Visual C++

Hello World: Array Addition

```
void AddArrays(int n, int * pA, int *
pB, int * pC)
{



   for (int i=0; i<n; i++)

   {
         pC[i] = pA[i] + pB[i];
   }

}
```

```
#include <amp.h>
using namespace concurrency;

void AddArrays(int n, int * pA, int *
pB, int * pC)
{
   array_view<int,1> a(n, pA);
   array_view<int,1> b(n, pB);
   array_view<int,1> sum(n, pC);

   parallel_for_each(
      sum.grid,
      [=](index<1> i) restrict(direct3d)
      {
         sum[i] = a[i] + b[i];
      }
    );
}
```

Basic Elements of C++ AMP code



```
void AddArrays(int n, int * pA, int * pB,
int * pSum)
{
    array_view<int,1> a(n, p
    array_view<int,1> b(n, p
    array_view<int,1> sum(n, pSum);

    parallel_for_each(
            sum.extent,
            [=](index<1> i) restrict(amp)
            {
                sum[i] = a[i] + b[i];
            }
}
```

parallel_for_each: execute the lambda on the accelerator once per thread

restrict(amp): tells the compiler to check that this code conforms to C++ AMP language restrictions

array_view: wraps the data to operate on the accelerator

extent: the number and shape of threads to execute the lambda

index: the thread ID that is running the lambda, used to index into data

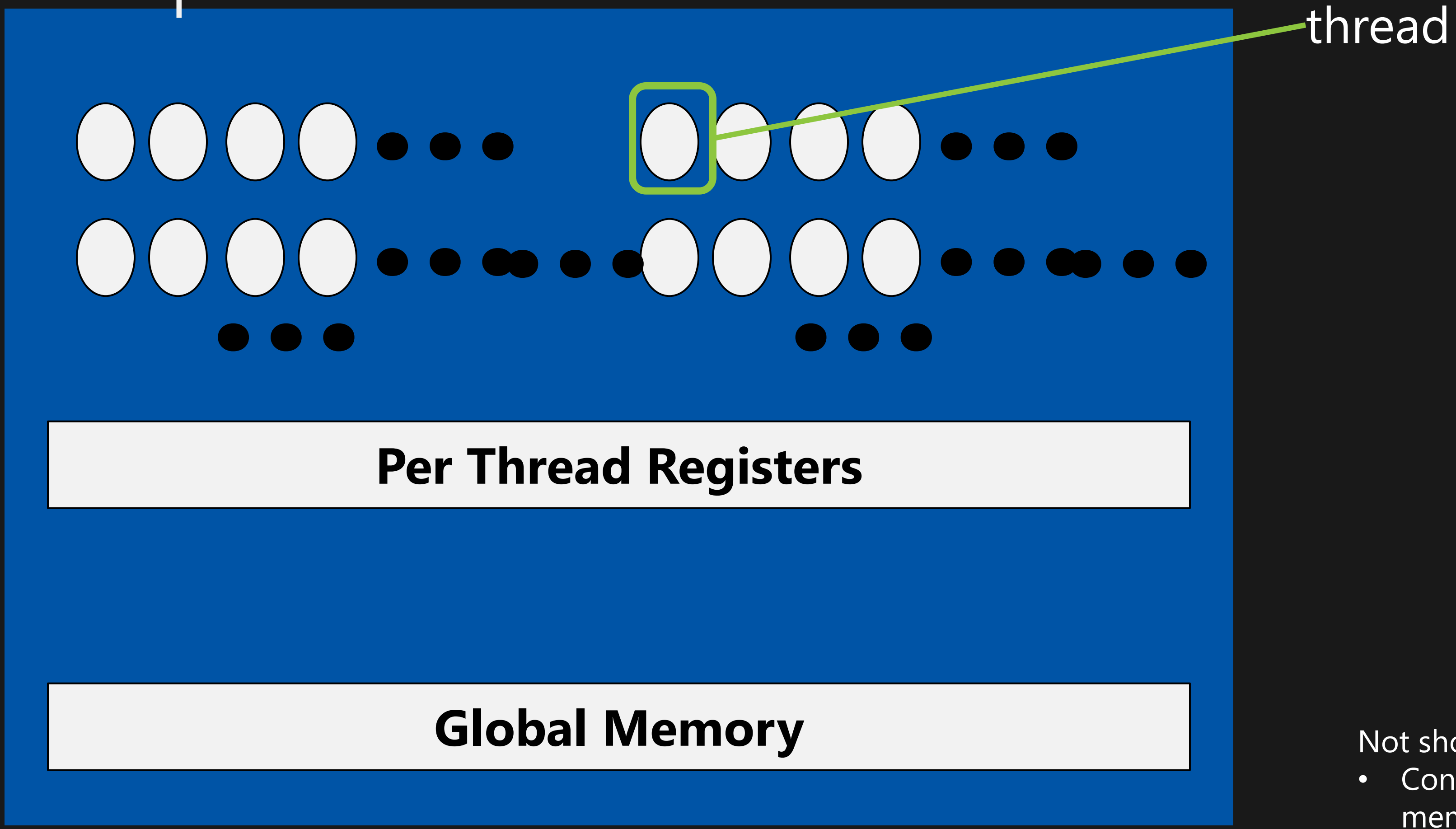
array_view variables captured and associated data copied to accelerator (on demand)

parallel_for_each

- Executes the kernel for each point in the extent
- As-if synchronous in terms of visible side-effects

```
parallel_for_each(
    e,   //e is of type extent<N>
    [ ](index<N> idx) restrict(amp)
    {
        // kernel code
    }
);
```

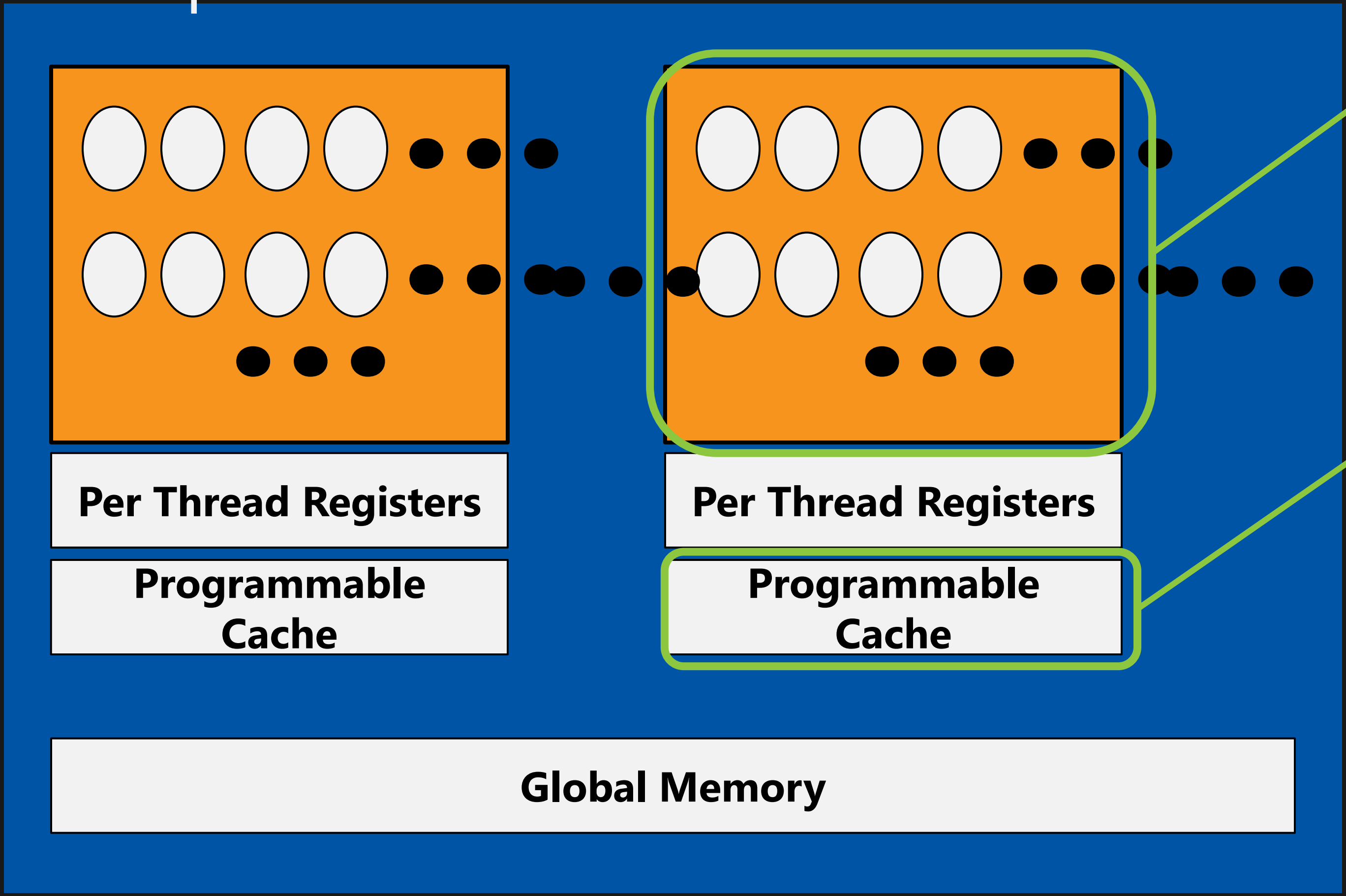
Hardware from a Developer Perspective



Not showing:

- Constant memory
- Memory controllers
- Schedulers
- Other caches
- Multi-GPU case

Hardware from a Developer Perspective



tile of threads

tile_static variables shared by threads in the same tile

Not showing:

- Constant memory
- Memory controllers
- Schedulers
- Other caches
- Multi-GPU case

parallel_for_each: tiled overload

- Schedule threads in tiles
  - Gain ability to use tile static memory
  - parallel_for_each overload for tiles accepts
    - tiled_extent<D0> or tiled_extent<D0, D1> or tiled_extent<D0, D1, D2>
    - a lambda which accepts
      - tiled_index<D0> or tiled_index<D0, D1> or tiled_index<D0, D1, D2>

```
array_view<int,1> data(12, my_data);

parallel_for_each(data.extent,
   [=] (index<1> idx) restrict(amp)
   { … });


parallel_for_each(data.extent.tile<6>(),
   [=] (tiled_index<6> t_idx)
restrict(amp)
   { … });
```

Demo

C++ AMP Parallel Debugger

- Well known Visual Studio debugging features
  - Launch (incl. remote), Attach, Break, Stepping, Breakpoints, DataTips
  - Toolwindows
    - Processes, Debug Output, Modules, Disassembly, Call Stack, Memory, Registers, Locals, Watch, Quick Watch
- New features (for both CPU and GPU)
  - Parallel Stacks window, Parallel Watch window, Barrier
- New GPU-specific
  - Emulator, GPU Threads window, race detection

Team oriented Features

- Code Review
- Unit Testing
- Code Coverage

Demo

Other IDE Enhancements

- Asynchronous Solution Load
- Graphics Tooling
- Windows 8 specific features
- XAML Designer
- Extension SDK
- …

Resources

Email: Sumit.Kumar@microsoft.com

MSDN Visual C++ Team Blog

- http://blogs.msdn.com/b/vcblog/

MSDN Visual Studio Team Blog

- http://blogs.msdn.com/b/visualstudio/

MSDN Native parallelism Team Blog

- http://blogs.msdn.com/b/nativeconcurrency/

Q&A