

# Christian Terboven terboven@rz.rwth-aachen.de

# Center for Computing and Communication RWTH Aachen University





PPCES 2010 March 22nd, RWTH Aachen University Agenda

MPI &

**OpenMP** 

Compilers

Debugging

**OpenMP** 

Debugging

MPI

Overview and Project Management

• The Microsoft and Intel compilers

Using MPI and OpenMP

Debugging OpenMP programs



2

• Debugging MPI programs

**Overview** 



### Visual Studio Versions overview

• As of today Visual Studio 2008 is the official product version and installed on <a href="mailto:cluster-win.rz.rwth-aachen.de">cluster-win.rz.rwth-aachen.de</a>.

- Runtime available on all compute node
- Intel C/C++ and Fortran compilers integration available
- Allinea DDTlight plug available
- Visual Studio 2010 is expected to be released on April 12th and currently installed as RC on <a href="mailto:cluster-win-lab.rz">cluster-win-lab.rz</a>.
  - Runtime available only on RZ\_HPCLAB compute nodes
  - No integration of Intel C/C++ and Fortran compilers
  - Comes with many enhancements for parallel programming

MPI &

OpenMP

Debugging

**OpenMP** 

Debugging

MPI

- Will be installed on cluster-win soon after release
- This course will cover VS2010 already (where possible)

Compilers



3

Center for

Overview

# VS2010 Overview (1/2)

Introduction into using Visual Studio 2008

- Only text-mode programs are considered here, as HPC applications typically do not use GUIs
- VS2010 offers great support for GUI development on Windows
- VS2010 provides good support for Parallel Programming:
  - Support for OpenMP for Shared-Memory parallel compilation
  - Debugging of parallel programs: OpenMP and MPI
  - Architecture-specific compiler optimizations

Compilers

Overview



4

 The first start of VS2010 may take a few minutes as the help system has to be updated. You are prompted to choose from several pre-defined sets of VS2010 configuration options: I nowadays choose the C# settings (previously: C++).

MPI &

OpenMP

Debugging

**OpenMP** 

Debugging

MPI





# Visual Studio: Project Management (1/5)

- Everything that you do in Visual Studio will take place within the context of a *Solution*.
  - A Solution is a higher-level container for other items, for example a *Project*. Any other kind of file type can also be added to a Solution, for example documentation items.
  - A Solution can not contain another Solution.
  - Solutions group and apply properties across projects.
- A *Project* maps one to one with a compiler target.

Compilers

A Project organizes the code.

Overview



6

• To start your work, a new Project has to be created with *File*  $\rightarrow$  *New*  $\rightarrow$  *Project...* 

MPI &

**OpenMP** 

Debugging

**OpenMP** 

Debugging

MPI





₩	Visual Stu	idio for HPC Appl	ication Developm Jdio: Proje	ent ect Managen	22.03.2010 – C. Terboven nent (3/5)
HAAC	Win32 Application Wizard - tes Welcome Overview Application Settings	e to the Win32 Application Wizard These are the current project settings: Click Finish from any window to accept the current setting After you create the project, see the project's readme.txt about the project features and files that are generated.	Choose have so	e <i>Empty project</i> ource files.	if you already
			Win32 Application Wizard -	test01 cation Settings	<u>? X</u>
		<previous next=""> Fir Next&gt; Compared to the second se</previous>	Overview Application Settings	Application type: C <u>W</u> indows application C <u>O</u> nsole application C <u>D</u> LL C <u>S</u> tatic library Additional options: Empty project Export symbols Precompiled header	Add common header files for:
WinHP <sup>3</sup> C 8					
<b>RNNTH</b> AACHEN UNIVERSITY	Center for Computing and Communication			< Previous	Next > Finish Cancel

O WinHP<sup>3</sup>C

9

Visual Studio for HPC Application Development

# Visual Studio: Project Management (4/5)

 An issue specific to our Cluster: The IntelliSense database may not be stored on a network drive. VS2010 resolves this automatically for you by selecting *Ok*.





22.03.2010 – C. Terboven

# Visual Studio: Project Management (5/5)



Visual Studio for HPC Application Development

In many cases, the shortest way to a desired operation can be found by right-clicking on a GUI element and using the context menu.

- Adding existing source code items (files) to a project: right-click on the Project (not the Solution !) and  $Add \rightarrow Existing Item...$
- Adding new items: right-click on the Project and Add  $\rightarrow$  New Item...
- The folders (e.g. Source Files) do not have any other meaning than aiding you in structuring the files in a project. They do not map to physical folders. Creating your own folders may help to organize large projects.



10

Center for

Computing and

Overview

Compilers

MPI & OpenMP Debugging **OpenMP** 

Debugging

MPI

HENE

11

### Source navigation in Visual Studio 2010 (1/2)

	-					
	<b>00</b> p	pi - Microsoft Visual Studio				
	File	Edit View Project Build Debug Team Data Tools Architecture Test Anal	yze Window Help			
	18	🗊 🕶 🕶 🚰 🛃 🎒   🐰 🗈 🛍 🔊 🗸 (** 🗸 📮 🗳 Debug 🔹	<b>29</b>			
		] 秘 법 🗠 偱   幸 幸   글 일   🗆 🖓 🤜 🖓 📮 👘 🖉 😓 🔍 🖕				
	×				-	
	đ		ar ** argy)			
	ollo	16 double fLocalPi:	Rank. int iNumProc)		· · ·	
		17 /* MPI Initialization */ ■♥ CalcPi(intn, intif	Rank, int iNumProcs)			
		18 MPI_Init(&argc, &argv); 10 MDI_Comm_cize(MDI_COMM_blopLDSiNumF =♦ f(double a)				
		20 MPI_Comm_rank(MPI_COMM_WORLD, &iMyRa @main(int argc, ch	ar ** argv)			
		<pre>22 char strHostname[MP1_MAX_PROCESSOR_NAME]; 23 int iDummy;</pre>				
		<pre>24 MPI_Get_processor_name(strHostname, &amp;iDummy);</pre>				
		<pre>25 std::cout &lt;&lt; "Process on " &lt;&lt; strHostname &lt;&lt; " runn 26 std::fflush(stdout):</pre>	ning." << std::e	ndl	;	
			/* the calculat	ion lcPi	is done here*/	Procel
	0	Selecting a scope + function to		5	Create Unit Tests	
		novigoto right into it's implo	fLocalPi = fPi	<b>-</b>	Go To Definition	F12
		havigate right into it's imple-	MPI_Reduce(&) Lo	→≣	Go To Declaration	Ctrl+F12
		mentation	double fTimeEnd		Find All References	Ctrl+K, R
			if (iMyRank ==	R.	View Call Hierarchy	Ctrl+K, Ctrl+T
			{		Go To Header File	
	0	Right-clicking a symbol opens up	cout << "\r << "Fr		Breakpoint	•
		a corresponding contact manue	<< "wa	≯≣	Run To Cursor	Ctrl+F10
		a corresponding context menu:	}	×	Cut	Ctrl+X
					Сору	Ctrl+C
	7	Center for MPI	or List ң Call Hierar		Paste	Ctrl+V
ITY		Communication Overview Compliers Open			Outlining	•
					-	





Visual Studio for HPC Application Development

### **Directory layout of Visual Studio solutions**

• The executable is created in the directory of the active configuration during the build process.

• Directory structure of a solution:

<top level> Given user directory

<project name> Created by VS2005 / VS2008

Debug Configuration: *Debug* 

Release Configuration: *Release* 

Platform: x64 (64bit for Amd64/Intel64)

Configuration: Debug

Configuration: Release



Overview

x64

Debug

Release

Compilers

MPI & OpenMP Debugging OpenMP

Debugging

MPI

### Visual Studio 2010: Performance Analyzer

 VS2010 comes with new tools to analyze your (parallel) application's performance: Analyze → Profiler → New Performance Session, then Analyze → Launch Performance

What method of profiling would you like to use?

OPU Sampling (recommended) Measures CPU-bound applications with low overhead

Compilers

- Instrumentation Measures function call counts and timing
- .NET Memory Allocation (Sampling) Track managed memory allocation
- Concurrency Detect threads waiting for other threads
  - ✓ Collect resource contention data
  - Collect thread execution data

All this can be done on the local Workstation, or in the Cluster!

Debugging

**OpenMP** 

Debugging

MPI

### • CPU Sampling:

**Overview** 

Wizard

 Will run the application under the control of a sampling performance analyzer (snapshots of the program's call tree are taken at regular intervals → non-intrusive, low overhead)

MPI &

**OpenMP** 



'inHP<sup>3</sup>C

22.03.2010 - C. Terboven

٢

ē

À

٨

4

----

1

đ

物

Debugging

MPI

# Performance Analyzer: CPU Sampling (1/2)

# Summary View highlights the program's Critical Path: Sample Profiling Report

2,702 total samples collected



#### Hot Path



15

The	most	expensive	call	nath	based	on	samp	le	counts	
THE	most	expensive	Call	paur	Daseu	011	samp		counts	

Overview

	Name	Inclusive %	Exclusive %
	≱ jacobi_aut.exe	100,00	0,00
	↓ _mainCRTStartup	100,00	0,00
Ċ	↓tmainCRTStartup	100,00	0,00
	▶_main	100,00	0,00
	🔶 _Jacobi	98,70	98,48

#### Related Views: Call Tree Functions



Compilers

MPI & OpenMP Debugging OpenMP



11.4 %

1.0 % 0.7 %

< 0.1 %

Center for Computing and

Communication

Visual Studio for HPC Application Development

#### 22.03.2010 – C. Terboven

### Performance Analyzer: CPU Sampling (2/2)

•

100.0%





#### Cost Distribution

Performance metric:

**Calling Functions** 

tmainCRTStartup

The cost distribution for the function and functions it calls is shown below.

Inclusive Samples %



16

100.0% main Total: Function Body < 0.1% 98.7% Jacobi InitializeMatrix CheckError 0.4% 0.3% Finish \_Init 0.1%





<u>Z</u> 2	Visual Studio for HPC Application Development	22.03.2010 – C. Terboven
25	Concurrency (2/3)	
QX	Concurrency Visualization	Visible Timeline Profile
	Visualizes the behavior of a multi-threated application	98% Execution
		1% Synchronization
		0% I/O
	CPU Utilization Threads Cores	0% Sleep
		0% Memory Management
		0% Preemption
		07% Of Processing
		Per Thread Summary
		File Operations
Sort by: Start Time		Demysury
Sort by: Start Time	p 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 3	32 34 36 38 40 42 44
Name Sec	nds	<u></u>
Disk 0 Reads		
Disk 0 Writes		
Main Thread(4108)		
Worker Thread(4696)		
Worker Thread(4684)		
Worker Thread(4672)		
Worker Thread(4620)		
Worker Thread(4448)		
Worker Thread(4740)		
Worker Thread(4736)		
Worker Thread(4732)		
Worker Thread(4720)		
Worker Thread(4716)		
Worker Thread(4724)		•

Visual St	tudio for HPC /	Application	Development	22.03.2010 – C. Terboven
		Сс	oncurrency (3	3/3)
<u> </u>				
Concurrence Visualizes the be	cy Visualization ehavior of a multi-threaded appli	cation		
	Utilization Threa	ds Co	res	
ĘS				
CPU Utilization   Threads   Cor	es			Demystify
Context switches that also cross from o	one logical core to another can	reduce the performance	of your process.	
Zoom				
Name Seconds	2 4 6 8	10 12 14		
Logical Core 3				
Logical Core 4				
Logical Core 5				
Logical Core 6				
Logical Core 7				
Logical Core 8				
Logical Core 9				
Logical Core 10				
Logical Core 11				
Thread Name	Cross-Core Context Switches	Total Context Switches	Percent of Context Switches that Cross Cores	
Worker Thread(4768)	272	808	33,66%	
Worker Thread(4708)	256	998	25,65%	
Worker Thread(4712)	244	979	24,92%	
Worker Thread(4620)	227	667	34,03%	
Worker Thread(4736)	220	938	23,45%	
Worker Thread(4672)	210	1 229	16 40%	

## Performance Analyzer: Thread Contention (1/2)

#### If threads compete for resources, they can get stalled: Ο

#### Most Contended Resources

Resources with the highest number of total contentions



22.03.2010 – C. Terboven

### Performance Analyzer: Thread Contention (2/2)

### • Reason for this contention: OpenMP Barrier

← ⇒ Current View: Call Tree		- 🪸 🐱	5 🖬 🙇 🖏 🖏 👌 i	1 🍇 🔏	
Function Name	Module Name	Inclusive Contentions	Exclusive Contentions	Inclusive Blocked Time	Exclusive Blocked Time
// jacobi_aut.exe		97	0	1.518	0
Jacobi\$omp\$1	jacobi_aut.exe	84	0	1.346	0
vcomp_barrier	VCOMP90D.DLL	84	84	1.346	1.346
_mainCRTStartup	jacobi_aut.exe	13	0	172	0
<ul> <li>Unknown Frame(s)</li> </ul>	UNKNOWN	13	0	172	0
Jacobi\$omp\$1	jacobi_aut.exe	13	0	172	0
_vcomp_barrier	VCOMP90D.DLL	13	13	172	172

Support for OpenMP constructs is not yet optimal

Compilers

**Overview** 

• This analysis is crucial if you do your own synchronization!

MPI &

**OpenMP** 

Debugging

**OpenMP** 

Debugging

MPI







### Performance Analyzer: More future features...

 Performance tuning can be a never ending story, so you need metrics to decide where to work / when to stop: Hardware Counter Information.

. . .

MPI &

**OpenMP** 

Compilers



**Overview** 

L2 information can be used to measure the memory bandwidth consumed by the application  $\rightarrow$  is your scalability limited by the system architecture? The FLOPS rate is good to estimate how efficient the code runs!

Debugging

**OpenMP** 

Debugging

MPI



/inHP<sup>3</sup>C

Agenda

MPI &

**OpenMP** 

Compilers

Debugging

**OpenMP** 

Debugging

MPI

Overview and Project Management

• The Microsoft and Intel compilers

Using MPI and OpenMP

Debugging OpenMP programs



23

• Debugging MPI programs

**Overview** 



Visual Studio for HPC Application Development 22.03.2010 – C. Terboven

# Visual Studio Configurations (1/3)

The set of compiler options is managed in a Configuration.

- There are two configurations pre-defined: *Debug* and *Release*.
  - Debug: typical options for debugging, no optimization.
  - Release: debugging still possible, some optimization options.
- The compile process can be triggered by right-clicking on the project and choosing *Build*. Or from the menu: *Build* → *Build* <projectname>.
- Build  $\rightarrow$  Build Solution builds all projects in the solution.
- During and after the compile process compiler output (informational messages, warnings, errors) is displayed in the tool windows *Output* or *Error List*.

**Compilers** 

• By double-clicking on such a message, the cursor jumps to the corresponding place in the code.

MPI &

OpenMP

Debugging

**OpenMP** 

Debugging

MPI



24

Center for

Overview





Visual Studio for HPC Application Development

22.03.2010 – C. Terboven

OK

Cancel

Apply

### Visual Studio Configurations (2/3)

 Right-clicking on a project and choosing Properties leads to the project configuration dialog.

pi_mpi Property Pages		<u> </u>
Configuration: Active(Debug)	▼ Platform: Active(Win32)	Configuration Manager
Configuration: Active(Debug) Cur. Configuration General Debugging VC++ Directories C/C++ Linker Manifest Tool XML Document Generator Browse Information Build Events Custom Build Step Code Analysis	Platform: Active(Win32)     Platform: Active(Win32)     Cur. Platform     Intermediate Directory     Target Name     Target Extension     Extensions to Delete on Clean     Build Log File     Platform Toolset     Project Defaults     Configuration Type     Use of ATL     Character Set     Common Language Runtime Support     Whole Program Optimization	Configuration Manager  \$(SolutionDir)\$(Configuration)\  \$(Configuration)\  \$(ProjectName) .exe  *.cdf;*.cache;*.obj;*.ilk;*.resources;*.tlb;*.tli;*.tlh;*.  \$(IntDir)\\$(MSBuildProjectName).log v100  Application (.exe) Use Standard Windows Libraries Not Using ATL Use Unicode Character Set No Common Language Runtime Support No Whole Program Optimization
	Output Directory Specifies a relative path to the output file directo	ory; can include environment variables.

22.03.2010 – C. Terboven

### Visual Studio Configurations (3/3)

### Build $\rightarrow$ Configuration Manager:

Configuration Manager		<mark>?</mark> ≥ Only	Win32 and x64
Active solution configuration:	Active solution platform:	are su	upported on or
Debug	Win32	I 1 0 0 0	
Debug	r deploy):	Cluste	er, not Itanium
<new></new>	Platform	Build	,
<edit></edit>	win32	▼ ▼	
	Configuration Manager		?
	Active solution configuration:	Active solution platform:	
	Release	<new></new>	
	Project contexts (check the project configuration	ons to build or deploy):	
	Project New Solution	Platform 2	Build
	- c-ser		
	Type or select	: the new <u>p</u> latform:	
	- x64	<b></b>	I
	Copy settings	from:	
	Win32		
	Create ne	w project platforms	
		OK Cancel	
You can create			
your own	3		Close
configurations			



Configuration: Active(Debug) General Debugging VC++ Directories □ C/C++ General Optimization Preprocessor Code Generation Language Precompiled Headers Output Files Browse Information Advanced Command Line

> E Linker General Input Manifest File Debugging System Optimization Embedded IDL Advanced Command Line

Important General Settings:

Microsoft C/C++-specific settings

- C/C++  $\rightarrow$  General
  - Addition Include Directories: Include Path

22.03.2010 – C. Terboven

- Linker  $\rightarrow$  General
  - Additional Library Directories: Library Path
- Linker  $\rightarrow$  Input
  - Additional Dependencies: Libraries to be used
- Important Optimization Settings:
  - $C/C++ \rightarrow Optimization$ 
    - Optimization: General Optimization Level
    - Inline Function Expansion: Inlining
  - $C/C++ \rightarrow$  Code Generation
    - Enable Enhanced Instruction Set: Vectorization



Overview

Compilers

MPI & **OpenMP**  Debugging **OpenMP** 

Debugging

MPI

### Using the Intel C/C++ compiler

- Create a (Win32) project using the Microsoft C/C++ compiler
- 2. Right-click on the Solution or on the Project
- 3. Intel Parallel Studio Installed (cluster-win-beta + -lab): Intel Parallel Composer → Use Intel C++

MPI &

OpenMP

Debugging

**OpenMP** 

Debugging

MPI

- 4. Intel Parallel Studio Not Installed (cluster-win): Use Intel(R) C++
- Solution or Project can be converted back to use Microsoft C/C++ as well.

**Overview** 

Currently only supported in Visual Studio 2008!

Compilers



# Intel C/C++-specific settings

 $\circ$  Most settings are the same as with Microsoft C/C++, but:

- Additional Optimization Settings: Ο
  - $C/C++ \rightarrow Optimization$ 
    - Generate Alternate Code Path and/or Use Intel(R) Processor **Extensions: Optimization for specific CPU**
    - Parallelization: Ask the compiler for automatic parallelization
  - $C/C++ \rightarrow Language$

Overview

- Recognize the Restrict Keyword: Enable C99 restrict (Tuning!)
- Enable C++0x Support: Enable first C++0x features

**Compilers** 

- Add to C/C++  $\rightarrow$  Command Line and Linker  $\rightarrow$  Command Line:
  - /Qtcheck: Enable source instrumentation for Thread Checker
  - /Qtprofile: Enable source instrumentation for Thread Profiler

MPI &

OpenMP

Debugging

**OpenMP** 

Debugging

MPI





22.03.2010 – C. Terboven

# Using the Intel Fortran compiler

- Open the Project Creation dialog via File  $\rightarrow$  New  $\rightarrow$  Project... 1.
- Select Intel(R) Fortran as project type 2.

····,···						
Project types:			<u>T</u> emplates:		.NET Framework 3.5	<b>→</b> 55 8:
Visual C++			Visual Studio installed t	emplates		
ATL				<b>—</b> 1		
CLR			Empty Project	Main	Program Code	
General			My Templater			
MFC			riy relipiates			
Smart Dev	lice		Search Online Templates			
Test						
Win32						
Intel(R) Fortra	an					
Console A	pplication					
Library						
QuickWin	Application					
Windowin	g Application					
COM Serv	er					
Database Proj	jects					
Other Langua	nes					
A project for creat	ting a command	-line app	plication			
Name:	Console 1					
1 K						Preven
Location:	lu:/				<b>_</b>	prowse
Solution Name:	Console 1			Create dire	ctory for solution	
					01	

Debugging

**OpenMP** 

Debugging

MPI

/inHP<sup>3</sup>C

30

Typically select *Empty Project* as best-suited option 3.

**Compilers** 

**Overview** 

Remaining project handling does not differ from C/C++! Ο

MPI &

**OpenMP** 



# Portable Time Measurement (1/3)

- Porting applications from Unix to Windows (or the other way Ο around) can be quite hard ... but it was not for most user codes (HPC) we tried on Windows.
  - (1) The most common problem was time measurement as gettimeofday() is not available on Windows,
  - (2) followed by directory management issues where ,/' instead of  $, \$  had been used before.
- In most cases we attacked (2) using #ifdefs. Ο

**Compilers** 

Overview

- Handling (1) depends on the programming language: Ο
  - C++: We have written a version of double realtime() for Windows and Unix.
  - FORTRAN: As the library (defined along with the language) already provides time measurement facilities, we used these.

MPI &

OpenMP

Debugging

**OpenMP** 

Debugging

MPI



### Portable Time Measurement (2/3)

#### #ifdef WIN32

```
#include <Windows.h>
#define Li2Double(x) ((double)((x).HighPart) * 4.294967296E9 + \
    (double)((x).LowPart))
```

#### #else

```
#include <sys/time.h>
```

Overview

```
#include <time.h>
```

#### #endif

Center for

```
double realtime (void) {
#ifdef WIN32
LARGE_INTEGER time, freq;
double dtime, dfreq;
if (QueryPerformanceCounter(&time) == 0) { ... error ... }
if (QueryPerformanceFrequency(&freq) == 0) { ... error ... }
return Li2Double(time) / Li2Double(freq);
#else
struct timeval tv;
gettimeofday(&tv, (struct timezone*)0);
```

Compilers

```
return ((double)tv.tv_sec + (double)tv.tv_usec / 1000000.0 );
```

MPI &

OpenMP

Debugging

OpenMP

Debugging

MPI



22.03.2010 – C. Terboven

Debugging

MPI

### Portable Time Measurement (3/3)

### • Taking time the MPI way:

#include <mpi.h>

```
double t1, t2, elapsed_seconds;
t1 = MPI_Wtime();
...
t2 = MPI_Wtime();
elapsed_seconds = t2 - t1;
```

### Taking time the OpenMP way:

```
#include <omp.h>
```







Agenda

MPI &

**OpenMP** 

Debugging

**OpenMP** 

Debugging

MPI

Overview and Project Management

• The Microsoft and Intel compilers

Using MPI and OpenMP

Debugging OpenMP programs



34

• Debugging MPI programs

**Overview** 

Compilers



Visual Studio for HPC Application Development

22.03.2010 – C. Terboven

## Enabling OpenMP (1/3)

### • OpenMP support has to be enabled in a configuration:

	Disable Language Extensions	No	• VS2005 C/C++
Configuration Properties	Default Char Unsigned	No	
General	Treat wchar_t as Built-in Type	Yes	
Debugging	Force Conformance In For Loop Scope	Yes	• VS2008 C/C++
⊡ C/C++	Enable Run-Time Type Info	Yes	
General Optimization Preprocessor Code Generation Language	OpenMP Support	Yes (/openmp) 💽	• VS2010 C/C++
Precompiled Headers Output Files Browse Information Advanced Command Line			OpenMP 3.0:
⊕- Linker ⊕- Manifest Tool ⊕- XML Document Generator			• Intel C/C++
⊕- Browse Information ⊕- Build Events ⊕- Custom Build Step			Intel FORTRAN
⊞- Web Deployment	OpenMP Support Enable OpenMP 2.0 language extensions.	(/openmp) OK Cancel Apply	



22.03.2010 – C. Terboven

## Enabling OpenMP (2/3)

### Known problem with Visual Studio and OpenMP:

c-omp.ex	e - Unable To Locate Component 🛛 🛛 🗙
8	This application has failed to start because vcompd.dll was not found. Re-installing the application may fix this problem.
	ОК

 The message appears if an OpenMP program has been compiled with OpenMP support enabled, but omp.h had not been included.



36

• Solution: include omp.h in at least one file per project.



pi\_mpi Property Pages



Configuration: Active(Debug) Platform: Active(Win32) **-**Ŧ Configuration Manager... Debugger to launch: + Common Properties Configuration Properties Local Windows Debugger General Debugging Command \$(TargetPath) VC++ Directories Command Arguments + C/C++ Working Directory ? × Environment + Linker Attach H Manifest Tool OMP\_NUM\_THREADS=2 Debugger Type + XML Document Generator Environment Browse Information ande environmente Build Events + SQL Debugging F Custom Build Step Code Analysis Inherited values: 37 Inherit from parent or project defaults Macros>> Environment OK Cancel Specifies the environment for the MPI & Debugging Debugging Center for **Overview** Compilers Computing and **OpenMP** MPI **OpenMP** Communicatio

# Enabling OpenMP (3/3)

 Setting the number of threads for debugging of OpenMP programs: set environment variable OMP\_NUM\_THREADS.

? X

## Enabling MPI (1/2)

 As MPI is implemented by a library, an application includes a file containing the type and function declarations named mpi.h and has to be linked with that library.

• Modify the project properties (1/2):

- Include Path: C/C++  $\rightarrow$  General  $\rightarrow$  Additional Include Directories

MPI &

**OpenMP** 

Debugging

**OpenMP** 

Debugging

MPI

- MS-MPI 2008 on cluster-win / cluster-win-lab : C:\Program Files\Microsoft HPC Pack 2008 SDK\Include
- I-MPI on cluster-win: C:\Program Files (x86)\Intel\ICT\3.1\mpi\3.1\[ia32|em64t]\include

Compilers

**Overview** 





## Enabling MPI (2/2)

- Modify the project properties (2/2):
  - Library Path: Linker  $\rightarrow$  General  $\rightarrow$  Additional Library Directories
    - MS-MPI 2008 on cluster-win / cluster-win-lab : C:\Program Files\Microsoft HPC Pack 2008 SDK\Lib\[i386|amd64]
    - I-MPI on cluster-win: C:\Program Files (x86)\Intel\ICT\3.1\mpi\3.1\[ia32|em64t]\lib
- No significant performance difference, so our advise:
  - Use MS-MPI with Visual Studio MPI Debugger
  - Use I-MPI with Intel Thread Analyzer & Collector
  - Sometimes a program does not like a specific MPI, so it is always a good thing to have a second one available...



39

Center for Computing an Communication

Overview

Compilers

MPI & OpenMP

ip

Debugging OpenMP

Debugging

MPI

Agenda

MPI &

**OpenMP** 

Compilers

Debugging

**OpenMP** 

Debugging

MPI

Overview and Project Management

• The Microsoft and Intel compilers

Using MPI and OpenMP

Debugging OpenMP programs



40

• Debugging MPI programs

**Overview** 



# Debugging Basics (1/2)

• A breakpoint can be set by clicking in the grey area left of the line number. Clicking again removes the breakpoint.

- Right-clicking a breakpoint opens the context menu with the following functions
  - Disable a breakpoint (temporarily)
  - Set breakpoint trigger conditions
  - Trigger filter for selected threads or processes
  - Define actions to be executed when the breakpoint is triggered

MPI &

OpenMP

Debugging

**OpenMP** 

Debugging

MPI



41

- Just hold the mouse over a variable for a short moment to get the actual value displayed (*hover*). This is also possible for expression (with some limitations).
  - Clicking the *pin* laminates the variable display

**Compilers** 

Overview



ACHEN UNIVERSITY

Visual Studio for HPC Application Development

22.03.2010 – C. Terboven

# Debugging Basics (2/2)

During a debugging session, the actual program location is 0 marked by a yellow arrow. You can drag this arrow up/down.



Visual Studio for HPC Application Development 22.03.2010 – C. Terboven

# Debugging OpenMP Programs (1/4)

- Debugging of OpenMP applications in Visual Studio works with all compilers: the Microsoft C/C++ compiler, the Intel C/C++ compiler and the Intel Fortran compiler.
- Note: If you use one of the Intel compilers or VS21010 and start a program with n threads, you will see n+1 threads (one management thread).
- We advise you to compile without any optimization for debugging, that means use the pre-configured *Debug* configuration and just enable OpenMP.
- Control debugging:



43

- Start / Continue, Break, Stop, Restart
- 💠 🗐 🗐 🖕 »
- » Show next statement, Step Into, Step Over, Step Out





Compilers



Debugging OpenMP





Visual Studio for HPC Application Development 22.03.2010 – C. Terboven

# Debugging OpenMP Programs (3/4)

• For all threads, you can view the *local* and *shared* variables.

- The *Locals* register contains all variables of the current scope.
- The Autos register contains a set of interesting variables guessed by the compiler – remarkably good.
- Some limitations when using the Intel Fortran compiler:
  - The Autos register is empty, Locals is working fine.
  - One can not (at least sometimes) identify the management thread by the name – it is the one you get an error message of no source code being available if you select it ;-)
  - Sometimes expressions have to be updated because of (possible) compiler optimization.



Parallel Stacks





Debugging OpenMP Programs (4/4)

 The Parallel Stacks window is a new feature of VS2010 and in the menu under *Debug*  $\rightarrow$  *Windows*  $\rightarrow$  *Parallel Stacks*:

22.03.2010 – C. Terboven

▼ 🗖 ×

Debugging

MPI

Debugging

**OpenMP** 

# Allinea DDTlite Overview

DDTlite is a plugin for Visual Studio 2008 SP1 (and newer) Ο

- Mainly marketed for improved MPI debugging experience
- But also improves OpenMP debugging experience
- Adds the following features:

**Overview** 

Control processes and threads individually, grouped or together

MPI &

OpenMP

- Examine variable values per thread / process
- Parallel Stack View per thread / process
- Location View per thread / process
- Thread / process group management



47

For a trial version go to www.allinea.com

Compilers



	_	
	0	Selected Processes and Threads
		All: 8 threads <u>n Si</u> <u>n i</u> Show Threads <u>Hide threads in external code</u>
		Break when: Every selected thread 💌 reaches a breakpoint Select all 9 threads in program
1		02345678
		Compare variables per thread:
	0	Process: [9812] jacobi-vs2008 even - Thread: [9288] vcomp: Persistent The V Stack Frame: Jacobi-vs2008 evenJacobitomot
		Variable - Parallel View
		ji
		□ 1 (1 thread) □ Thread 0 □ 100 (1 thread) □ Thread 1
		Auto Update
		Selected Processes and Threads     Image: A Variable     I
		Solution Explorer - jacobi-vs2008 - 4 X main.c jacobi.c
<sup>1</sup> C		Image: Solution 'jacobi-vs2008' (1 project)       72         Image: Solution 'jacobi-vs2008' (1 project)       72         Image: Solution 'jacobi-vs2008       73         Image: Solution 'jacobi-vs2008       73         Image: Solution 'jacobi-vs2008       74         Image: Solution 'jacobi-vs2008       75         Image: Solution 'jacobi-vs2008       74         Image: Solution 'jacobi-vs2008       74         Image: Solution 'jacobi-vs2008       74         Image: Solution 'jacobi-vs2008       75         Image: Solution 'jacobi-vs2008       74         Image: Solution 'jacobi-vs2008       75         <
ΤY	Ce Co Co	Anter for omputing and ommunicationOverviewCompilersMPI & OpenMPDebugging OpenMPDebugging MPI

Debugging OpenMP Programs w/ DDTlite

22.03.2010 – C. Terboven

Agenda

MPI &

**OpenMP** 

Compilers

Debugging

**OpenMP** 

Debugging

**MPI** 

Overview and Project Management

• The Microsoft and Intel compilers

Using MPI and OpenMP

Debugging OpenMP programs



49

Debugging MPI programs

**Overview** 



# Debugging MPI programs (1/6)

- MS-MPI works best, but you should be able to use I-MPI as well. At least the following instructions work for both.
- Visual Studio supports debugging of MPI programs using the Cluster Debugger. As far as I know – or was able to verify – the cluster debugger only works with the Microsoft C/C++ compiler and not with projects using the Intel C/C++ compiler or the Intel FORTRAN compiler.
- In the project properties under *Debugging*, choose the *MPI* Ο *Cluster Debugger* as *Debugger to launch*. For VS2008 only:
  - MPIRun: "C:\Program Files\Microsoft HPC Pack 2008 SDK\Bin "

**Compilers** 

- MPIRun Arguments: for example -n 2
- **MPIShim Location:**

Overview

It is not possible to specify a path containing empty spaces here, so you have to copy MPIShim from c:\program files[ (x86)]\microsoft visual studio 9.0 \common7\ide\RemoteDebugger\x86[or x64]\MPIShim to a suitable location.

MPI &

OpenMP

Debugging

**OpenMP** 

Debugging

MPI





# Debugging MPI programs (2/6)

- In order to stop all processes at a breakpoint, please check for the following option: In *Tools* → *Options* → *Debugging* → *General* the checkbox *Break all processes when one process breaks* has to be activated.
- Select the current process using the *Processes* register.
- Problem: if one process does a step, all other processes make a step as well.
  - Solution 1: Be clever setting breakpoints ...

Compilers

 Solution 2: Detach the second (and third and fourth and ...) process from the VS debugger (remember the right-click and context menu), open up another VS instance (maybe more) and attach to the process.

MPI &

OpenMP

Debugging

**OpenMP** 

Debugging

MPI

- Solution 3: Use DDTlight

Overview



<b>HE</b>	Visual Studio for HPC Application Development 22.03.2010 – C. Terbover Debugging MPI programs (3/6)					
	O In pi_mpi Pro	VS2010 V	ou can just go w	vith the defaul	ts: ?×	
	Configurat	tion: Active(Debug)	Platform: Active(Win32)     Debugger to launch:	<u>.</u>	Configuration Manager	
	k ⊡ Conf ( [ 5 \ \ 2 ⊡ (	iguration Properties General Debugging /C++ Directories C/C++ inker	MPI Cluster Debugger           Run Environment           Deployment Directory           Working Directory	localhost/4		
Node Selector Head Node:		localhost	X	Debug on the lo workstation with	cal n	
Number of pro Schedule one Pick nodes fro	ocesses: process per: om:	4 	▼ ▼ ▼		ses	
Manually s	elect nodes to	include in the allocation	PU (MHz) Memory (MB) Cores	50000 Yes Yes		
				per of processes, and the allocation	n of processes to machines, if	
			OK Concel		Cancel Apply	
				OpenMP	OpenMP MPI	

22.03.2010 – C. Terboven

### Debugging MPI programs (4/6)

Master
Slave
Slave

Nod

Visual Studio for HPC Application Development

22.03.2010 – C. Terboven

Transport

Default

Default

Default

Default

Transport Qualifier

WINIHNC04

WINIHNC05

WINIHNC05

WINIHNC04

### Debugging MPI programs (5/6)

### • You can also F5 to the Cluster, if there are free slots:

e Selector					_ [			
						_		
lead Node:	WINWCC	01				•		
lumber of processes:	4					•		
chedule one process per:	Socket					<b>-</b>		
ick nodes from	Computet	lodes						
ick houes from.	Jeompater	voues						
Manually select nodes to	include in t	the allocation						
Node 🔺		CPU Usage	CPU (MHz)	Memory (MB)	Sockets	•		
WINHTC07		100%	2999	16383	2			
WINHTC08		99%	2999	16383	2			
WINHTC09		99%	2999	16383	2			
WINHTC10		77%	2999	16383	2			
WINHTC11		98%	2999	16383	2			
WINHTC12		80%	2999	16383	2			
WINHTC13		99%	2999	16383	2			
WINIHNC01		0%	2933	49141	2	•		
				ок	Cancel			
Processes								
🕨 m   🗺 💭	· 🖕 📃	🤹 🔜						
Name 1	ID Path	ı		Title		State		Debugg
pi_mpi.exe	3468 C:\U	Jsers\ct74776	4\AppD	pi_mpi		Paused (eve	ent queued)	Native
pi_mpi.exe	4608 C:\U	Jsers\ct74776	4\AppD	pi_mpi		Break		Native
pi_mpi.exe	4704 C:\U	Jsers\ct74776	4\AppD	pi_mpi		Break		Native
pi mpi.exe	5348 C:\U	Jsers\ct74776	4\AppD	pi mpi		Paused (eve	ent queued)	Native



Visual Studio for HPC Application Development

22.03.2010 – C. Terboven

### Debugging MPI programs (6/6)

### A VS2010 debug job running on our Cluster:

Active (1	)				
Filter: Job na	ame Owner	<ul> <li>Submit ti</li> </ul>	me 👻 Project name	• 🔻 🗙	
J Joi	b Name	State Owner	Priority Submit Time	Requeste	1
28628 Vis	sual Studio Debugging Session	n Running WIN-HPC\ct7	747764 Normal 13.03.2010 1	14:25:39 4-4 Sockets	
					Project name     V
					District Colored Trans
					Normal 13.03.2010 14:25:39 4-4 Sockets
i					
🖃 Job Na	ame : Visual Studio Debug	aina Session	Exp	and parametric tasks	3
			~ -+		
Task J	ob Details   Activity Log	1	1	1	
Task ID	Task Name	State Size	Command Line	Requested Resource:	
2	Visual Studio Debugging	Rupping	"\\WINWCC01\CepSpo	4-4 Sockets	
3	Delete Work Directory Fil	Queued	"%CCP HOME%\bin\mp	4-4 Sockets	
		•	- '		
		📄 Job Name : V	isual Studio Debugging Se	ssion	Expand parametric tasks
		Taala Lab Data	la Activity Log		
		12 02 2010 14-25	29. Crosted by WIN UPC\ct74	7764	
		13.03.2010 14:25	:39 Submitted	//04	
		13.03.2010 14:25	40 Started	10	
		13.03.2010 14:25	40 Started on WINIHNC04 with 40 Started on WINIHNC05 with	h 16 cores	
Center for Computing an		u Compil	MPI &	Debu	gging Debugging

### **DDTlite: Overview**

Allinea DDT Lite is an add-in for Visual Studio 2008 SP1

- Currently an additional patch to VS2008 is required
- Significantly improves the MPI debugging experience
  - Debug / Control MPI processes individually
  - Debug / Control groups of MPI processes individually

MPI &

**OpenMP** 

Debugging

**OpenMP** 

Debugging

MPI

- Display variable values per process side-by-side
- Display MPI process stacks side-by-side



56



Compilers

**Overview** 



### DDTlite: Usage

MPI &

**OpenMP** 

Debugging

**OpenMP** 

Debugging

MPI

Configure project to use the MPI Cluster Debugger

- Tools -> Add-in Manager
  - Select Allinea DDTlite
  - Enable Startup
- Start Debugging (e.g. via F5)
- Debug -> Windows -> Show DDTLite Windows

Compilers

- Selected Processes and Threads
- Groups Parallel View
- Variable Parallel View
- Location Parallel View
- Parallel Stack View

**Overview** 







The End

# Thank you for

### your attention!



58



22.03.2010 – C. Terboven