IWOMP 2015

11th International Workshop on OpenMP

in SuperC RWTH Aachen University Templergraben 57

Wednesday 30th September – Tutorial

Time	Presentation / Speaker	Track
09:00	*Welcome coffee and set-up for delegates in room*	Foyer - Generali
10:00	OpenMP Does Not Scale *Ruud van der Pas, Oracle Corporation* *	Track A - Generali
11:00	**Coffee Break**	Foyer
11:30	Advanced OpenMP Tasking *Michael Klemm, Intel Corp. and Christian Terboven, RWTH Aachen University* *	Track A - Generali
12:30	**Lunch**	Foyer
14:00	SIMD Programming with OpenMP *Georg Zitzlsberger, Intel Corp*oration *	Track A - Generali
15:00	**Coffee Break**	Foyer
15:30	OpenMP 4.x Accelerator Model (Part 1 + Part 2) *Eric Stotzer, Texas Instruments* *	Track A - Generali
17:30	*End of Tutorial*	

*Please find abstracts to tutorials on page 2

IWOMP 2015

11th International Workshop on OpenMP

in SuperC RWTH Aachen University Templergraben 57

abstracts:

Ruud: ***OpenMP Does Not Scale***

Unfortunately it is a very widespread myth that OpenMP Does Not Scale – a myth we intend to dispel in this talk. Every parallel system has its strengths and weaknesses. This is true for clustered systems, but also for shared memory parallel computers. While nobody in their right mind would consider sending one zillion single byte messages to a single node in a cluster, people do the equivalent in OpenMP and then blame the programming model.
Also, shared memory parallel systems have some specific features that one needs to be aware of. Few do though. In this talk we use real-life case studies based on actual applications to show why an application did not scale and what was done to change this. More often than not, a relatively simple modification, or even a system level setting, makes all the difference.

Christian / Michael: ***Advanced OpenMP Tasking***

OpenMP is a popular, portable, widely supported and easy-to-use shared-memory model. Developers usually find OpenMP easy to learn. However, they are sometimes disappointed with the performance and scalability of the resulting code. This disappointment stems not from shortcomings of OpenMP but rather with the lack of depth with which it is employed. This is particularly true for advanced language constructs, such as tasking. In this tutorial we give an in-depth introduction into the OpenMP task model. We focus on performance aspects and present best practices and when to use tasks and how to achieve the expected performance.

Georg: ***SIMD Programming with OpenMP***

If you think OpenMP is merely about threading then you might be interested in the latest features of OpenMP 4.x that exploit the SIMD capabilities of modern processors. Since processors tend to spend more die space for SIMD, growing with every new generation, the so-called “vectorization” becomes more important. Whereas threading is already covered well, vectorization is still is an underdog. In this tutorial we provide an introduction to vectorization extensions of OpenMP 4.0 and the upcoming version.

Simplified examples extracted from recent Intel Parallel Computing Center projects will be used as demonstration.

Attendees will get a set of different examples to become accustomed with the different vectorization techniques of the latest OpenMP standards.

Eric: ***OpenMP 4.x Accelerator Model***

OpenMP is the dominant programming model for shared-memory parallelism in C, C++ and Fortran due to its easy-to-use directive-based style, portability and broad support by compiler vendors. Compute-intensive application regions are increasingly being accelerated using specialized heterogeneous devices and a programming model with similar characteristics is needed here. This tutorial will focus on the OpenMP 4.0 accelerator model that provides such a programming model.
For this half-day tutorial we assume attendees have a basic understanding of OpenMP concepts. We quickly review OpenMP programming topics that are most relevant to the accelerator model. We focus on how the OpenMP execution and memory models were extended to support heterogeneous devices. We cover the new device constructs and API routines that were added in OpenMP 4.0, and we work through some example code. Finally, we preview some of the upcoming features coming in OpenMP 4.1.