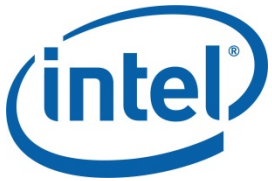
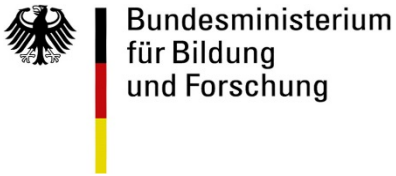
ELP

Effektive Laufzeitunterstützung für zukünftige Programmierstandards





Zwischenbericht

ELP

Speaker: Christian Terboven, RWTH Aachen University

Agenda

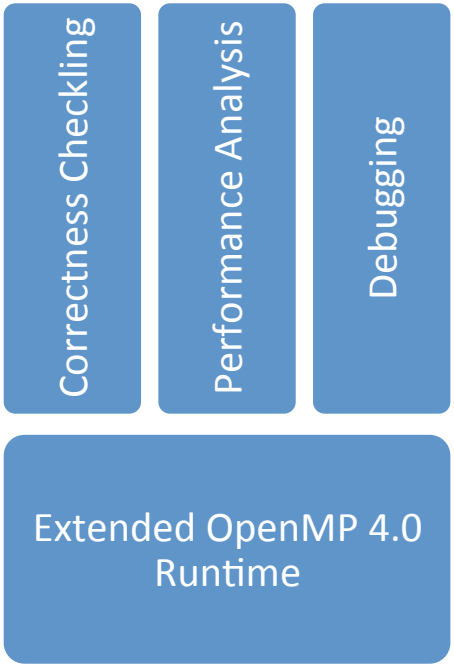
- **ELP Project Goals**
- **ELP Achievements**
- **Remaining Steps**

ELP Project Goals

- **Goals of ELP: Improve programmer productivity**
  - By influencing the Programming Standards
  - By extending Compiler / Runtime / Tools Support
  - Reducing workflow complexity
    - tight tool integration in the workflow
- **Make the correctness analysis part of a standard workflow**
- **Results for SMEs and General Dissemination**
  - Know-How in designing and operating accelerator-based HPC systems
  - Improved user support in code parallelization projects and in benchmarking

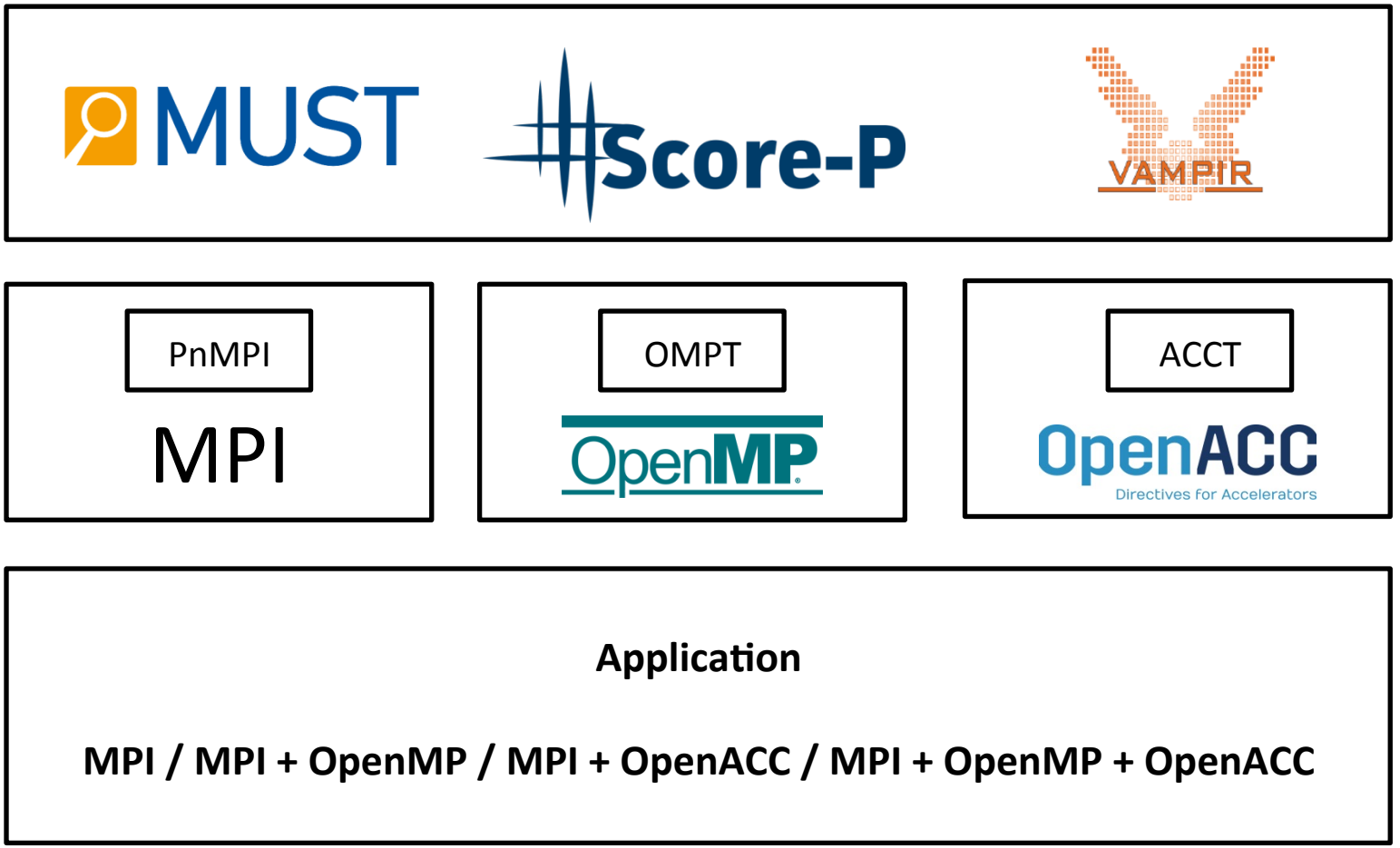
The ELP Approach for Programmer Productivity

- **Support for directive-based standards**
  - Focus on OpenMP 4.0
    - Infrastructure is available as open source
    - OpenACC support is limited to selected activities
  - Extended OpenMP Runtime
    - OpenMP standardization work
    - prototype implementation of proposed extensions, i.e. OMPT in Intel runtime
- **Performance Analysis on accelerators with Score-P and Vampir**
- **Correctness Checking with MUST**
  - Support for Hybrid Programs in MUST: MPI + OpenMP
  - new kinds of races, support of new APIs
  - hybrid deadlocks



Intel Open Source

Tools Landscape



MUST and Score-P

- **MUST was limited on MPI correctness analysis:**
  - Type mis-matches, overlapping of buffer usage, …
  - Deadlocks resulting from MPI calls
- **ELP: support for hybrid MPI + OpenMP programs**
  - Pure OpenMP 4.0: data races between host and accelerator, deadlocks, …
  - Hybrid: MPI deadlocks involving threads, data races involving data transfer, ...
- **Score-P: Scalable performance measurement infrastructure for parallel codes**
  - Measurement infrastructure for profiling, event tracing and online analysis of applications
- **ELP: Support for OpenMP 4.0 and OpenACC programs**
  - Standardized tools interfaces: OMPT and ACCT
  - Recording of
    - OpenMP target Events
    - OpenMP Events on the target device (offloaded via the target construct)
    - OpenACC Events

Agenda

- ELP Project Goals
- **ELP Achievements**
- Remaining Steps

Project Achievements: Overview

- **MUST + GTI infrastructure is ready for hybrid parallel programs**
  - Set of new checks: correct use of OpenMP barriers, simple hybrid deadlocks
- **OMPT completed and ported to latest open source OpenMP Runtime**
  - Patch provided for LLVM Runtime, close collaboration with RICE and UOREGON and INTEL
- **Proposal for an OMPT extension to support OpenMP 4.0 constructs, including a prototype implementation in the Intel/LLVM OpenMP Runtime**
  - supporting the development of OpenMP and OMPT: OpenMP 4.0 target
- **OpenACC Profiling Interface is part of OpenACC 2.5**
  - supporting the development of OpenACC
- **8 peer-reviewed papers published**

Project Achievements: OMPT (1/2)

- **Contribution to future OpenMP specification (technical report)**
  - Proposal for an extensions of the OpenMP Tools Interface [1]
  - Revised technical report for an OpenMP Tools Interface [2]
  - Adding an interface for buffer traces (collaboration with RICE)
    - Asynchronous buffer handling
    - Approach similar to CUDA performance tools interface (CUPTI)
    - Reference implementation available [3]

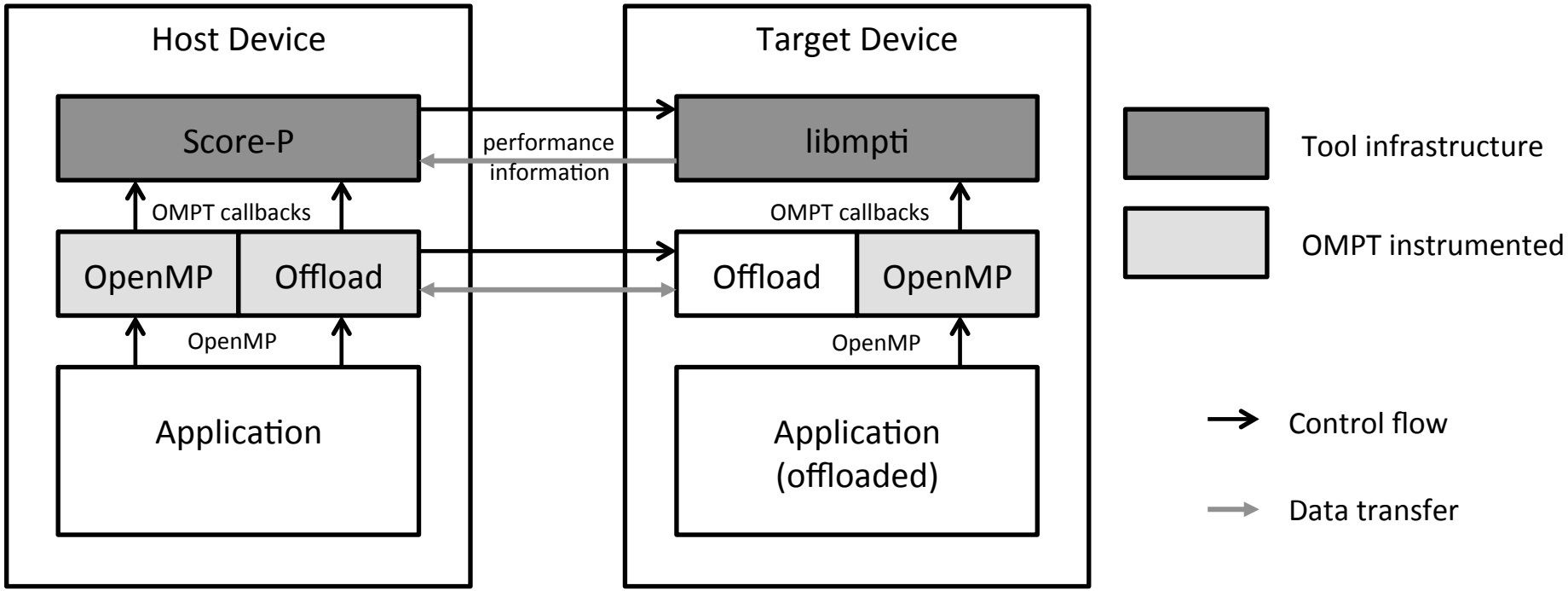
[1] Tim Cramer, Robert Dietrich, Christian Terboven, Matthias S. Müller, Wolfgang E. Nagel: **Performance Analysis for Target Devices with the OpenMP Tools Interface**. In Proceedings of 20th IEEE International Workshop on High-level Parallel Programming Models and Supportive Environments (HIPS), pages 216-224. 2015, Hyderabad, May 2015.

[2] Alexandre E. Eichenberger, John M. Mellor-Crummey, Martin Schulz, Nawal Copty, Jim Cownie, Tim Cramer, Robert Dietrich, Xu Liu, Eugene Loh, and Daniel Lorenz et al. **OMPT: An OpenMP Tools Application Programming Interface for Performance Analysis.** Revised OpenMP Technical Report 2, https://github.com/OpenMPToolsInterface/OMPT-Technical-Report, August 2015.

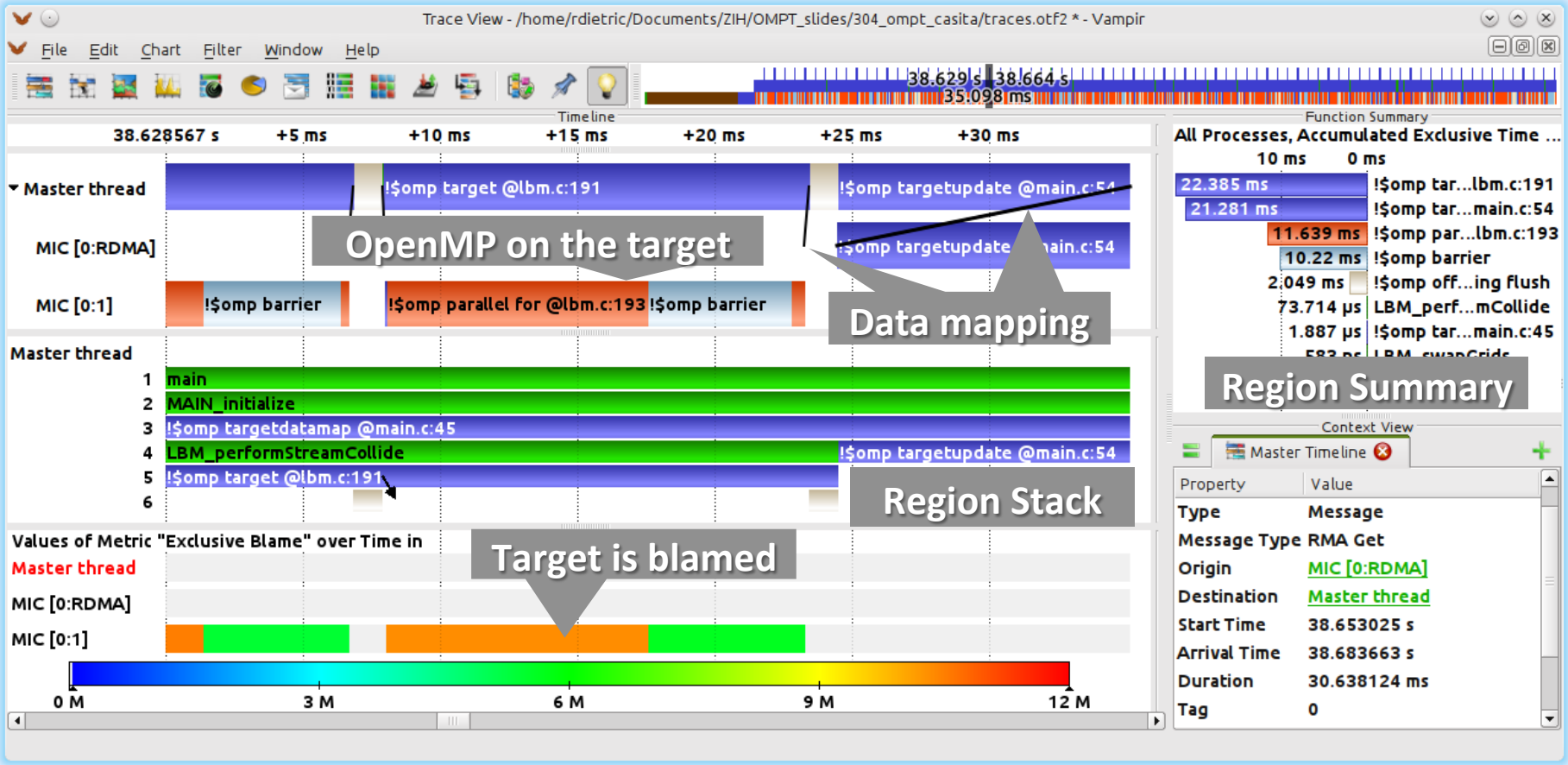
[3] https://github.com/OpenMPToolsInterface/LLVM-openmp, October 2015

Project Achievements: OMPT (2/2)

- **OMPT extension in Score-P**
  - Low overhead
    - 1.5 % (host-sided events only)
    - 12.9 % (host- and device-sided events)
  - Future Work: Use OMPT Buffering API



Vampir: Performance Data Visualization



Visualization of OTF2 traces that have been generated by Score-P

Screenshot shows prototype implementation for an OpenMP 4.0 program that performs computation on a target device

Project Achievements: OpenACC Tools Interface

- **OpenACC 2.5 standard released in November 2015**
- **Technical report has been incorporated in the official specification**
  - Chapter 5: Profiling Interface
  - Defines a set of OpenACC runtime events
- **Score-P prototype implementation**
  - Tested with the PGI 15.x compilers
  - Only a few PGI-specifics left
  - Validated with OpenACC benchmarks of the SPEC ACCEL suite
- **Dissemination**
  - Presented at SC'15, HPCwire article:
    „OpenACC Reviews Latest Developments and Future Plans"
  - Paper published at ICPP'15:
    "OpenACC Programs Examined: A Performance Analysis Approach"

Project Achievements: Hybrid Correctness Checking

- **OMPT: integration into MUST‘s event system completed**
  - Prototype OMPT 4.1 target support available
- **Detections already implemented in MUST, utilizing OMPT:**
  - Wrong threading level
    - With more than one thread *MPI_Init* is used instead of *MPI_Init_thread*
    - *MPI_THREAD_SINGLE* is used with multiple threads
    - MPI-Thread-Level < MPI_THREAD_MULTIPLE when multiple threads issue MPI-Calls concurrently
  - Multiple threads of a team passing different barriers
  - Usage of uninitialized locks
  - Deadlock using a single lock
  - Deadlock using multiple locks

Agenda

- ELP Project Goals
- ELP Achievements
- **Remaining Steps**

Next Steps

- **Continued quality control wrt/ compatibility w/ Linux SW stacks**
- **Completion of the memory access tracing**
  - will enable us to to extend the correctness checks to include data races
- **Score-P**
  - Enhance stability and functionality of the prototypes (as always ☺)
  - Advance standardization of OpenMP and OpenACC tools interfaces to enable the implementation into the stable Score-P product release
    - OpenACC part almost done, expected for next release
  - Integrate OMPT device activity buffering API
- **Review OpenMP 4.5 (released in Nov. 2015)**
  - Support extended target functionality
- **Support development of OpenMP 5.0 to include OMPT**

The ELP-Team Thanks You!

http://www.vi-hps.org/projects/elp/

Christian Terboven
terboven@itc.rwth-aachen.de