ELP
Effektive Laufzeitunterstützung für zukünftige Programmierstandards

Übersicht und Ziele
ELP

Speaker: J. Wender, C. Terboven
Agenda

- ELP Project Goals
- The MUST, Score-P and VAMPIR Tools
- ELP Achievements
- Next Steps
**ELP Project Goals**

- **Goals of ELP: use of compiler- and runtime-methods to**
  - Improve programmer productivity
    - correctness checking of hybrid programs
    - performance analysis on accelerators
  - Reduce workflow complexity
    - tight tool integration in the workflow
  - Support for directive-based standards
    - Focus on OpenMP 4.0 for now as all infrastructure is available in open source, OpenACC support is limited to selected activities
  - **Observation: directive-based standards gain popularity in the HPC community due to their productivity advantage**
Improved Productivity through Performance Analysis and Correctness Checking of OpenMP 4.0 Programs
- Performance analysis on accelerators based on Score-P
- Correctness: new kinds of races, support of new API, hybrid deadlocks
- Support for Hybrid Programs in MUST: MPI + OpenMP

Extended OpenMP Runtime
- OpenMP standardization work
- Prototype implementation of proposed extensions, i.e. OMPT in Intel runtime

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

And: make the correctness analysis part of a standard workflow
MUST: Correctness Checking of Parallel Programs

- **MUST: Marmot Umpire Scalable Tool, based on**
  - MARMOT (HLRS → TUD): tool for analyzing and checking MPI programs
  - UMPIRE (LLNL): monitors each call to the MPI library for incorrect/unsafe use

- **So far, MUST was limited on MPI correctness analysis:**
  - Usage of datatypes, groups and communicators, request handles, …
  - Memory leak checks, 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, …

Speaker: J. Wender, C. Terboven
Score-P: Performance Measurement

- Scalable performance measurement infrastructure for parallel codes
  - Measurement infrastructure for profiling, event tracing and online analysis of HPC applications
  - Used by Vampir, Periscope, Scalasca and TAU
- Existing parallel paradigm support in Score-P: MPI, OpenMP 3.0, ...
  - OPARI2 as instrumenter for OpenMP
  - Source-to-source instrumentation is error prone
- ELP: Support for OpenMP 4.0 and OpenACC programs
  - Standardized OpenMP tools interface (OMPT)
  - Standardized OpenACC performance tools interface (ACCT?)
  - Recording of OpenMP 4.0 target and OpenACC directives and OpenMP on the target device (offloaded via the target construct)
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

Speaker: J. Wender, C. Terboven
Project Achievements

- **MUST + GTI infrastructure is ready for hybrid parallel programs**
  - still supports MPI-only, but also supports OpenMP-only
  - MUST release 1.4.0, this version is now thread-safe

- **OMPT ported to the latest open source Intel OpenMP Runtime**
  - Version 14.0, 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 OpenMP Runtime**
  - supporting the development of OpenMP and OMPT: OpenMP 4.0 target
  - offering a x86-based open source implementation

- **First steps towards a Memory Tracing Tool based on Intel PIN**
TR14-2: OpenACC performance tools interface

Prototype of Score-P Support for the Intel Xeon Phi for the OpenMP 4.0 target construct

- Score-P adapter implemented
- Support for OpenMP 4.0 target programs on Intel Xeon Phi
  (Paper: “Performance Measurement for the OpenMP 4.0 Offloading Model”)
- Support for recording of OpenACC directives based on the PGI implementation of the upcoming OpenACC performance tools interface defined in TR14-2 (www.openacc.org)

4 papers published
Next Steps

- **Integration of OMPT callbacks into MUST**
  - will enable us to perform first correctness checks of hybrid parallel programs

- **Data compression in the memory access tracing**
  - will enable us to extend the correctness check to include data races

- **Score-P**
  - Enhance stability and functionality of the prototypes
  - Improve visualization of collected performance data
  - Advance standardization of OpenMP and OpenACC tools interfaces to enable the implementation into the stable Score-P product release