

Performance Analysis of OpenMP Target Offloading in Score-P

Authors: Jan André Reuter¹, William R. Williams², Bernd Mohr¹, **Contact:** j.reuter@fz-juelich.de

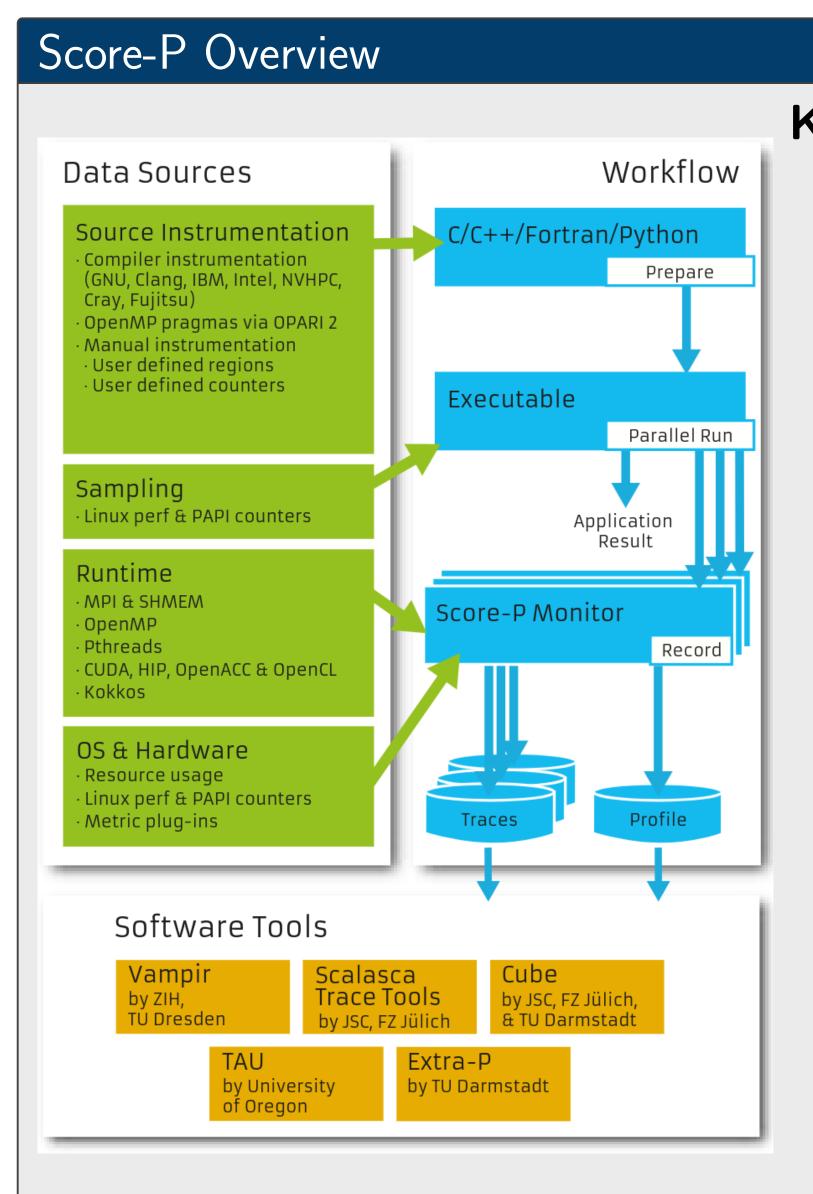
¹Jülich Supercomputing Centre (JSC), Forschungszentrum Jülich GmbH, Jülich, Germany

²Center for Interdisciplinary Digital Sciences (CIDS), Department Information Services and High Performance Computing (ZIH), TUD Dresden University of Technology, Dresden, Germany



Introduction

- OpenMP supports offloading to accelerators via target directives, with the option to target different architectures, selecting them at runtime, interoperability with low-level APIs, and more. In recent years, OpenMP offloading gained a growing user base with frameworks offering support as well.
- With the OpenMP Tools Interface, performance analysis of OpenMP target regions is possible. However, the current support in performance tools is limited. Here, we present a prototype for performance instrumentation and measurement of applications using OpenMP offloading with Score-P.



Key features:

- Highly scalable, supporting platforms $\geq 500 k$ cores
- Supported platforms:
 Linux clusters, HPE Cray,
 Fujitsu & more (using x86_64, Power & ARM)
- Supported parallelization models: MPI, SHMEM,
 OpenMP, Pthreads,
 CUDA, HIP, OpenACC,
 OpenCL & Kokkos
- Recording of I/O operations
- Profiling (CUBE4) & event tracing (OTF2)
- User library wrapping for C/C++ libraries
- Available as open source under a 3-clause BSD license

URL: https://www.score-p.org – **Contact:** support@score-p.org

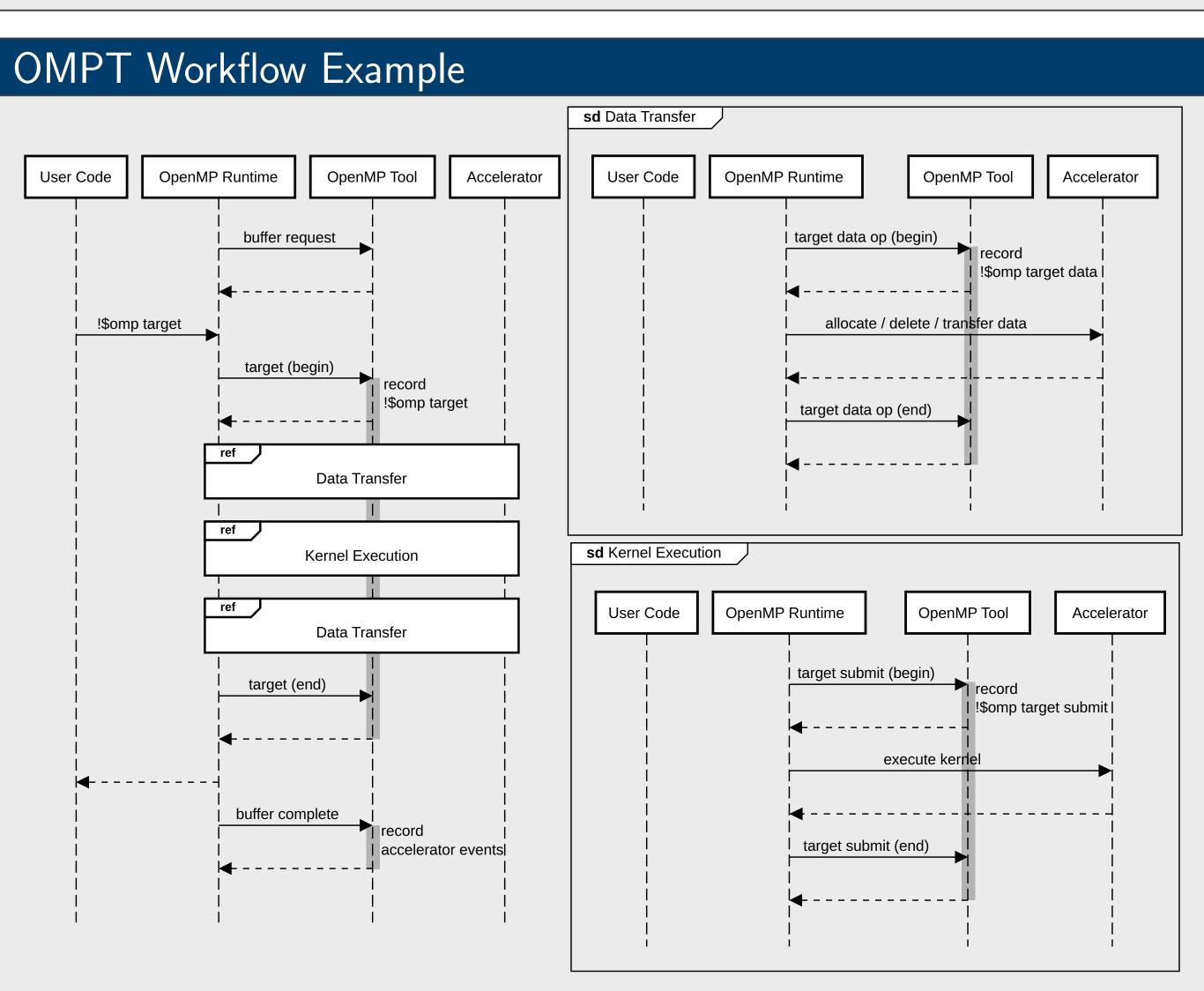
Current OMPT Compiler Vendor Support for Score-P

			л . г .
Compiler	Host	Events	Accelerator Events
	Host	Target	Device Tracing
AOMP 18.0-1	Full	Full	Full
CCE 17.0.0	None	Full	None
Clang 18.1.6	Full	Partial	None
GCC 14.1	None	None	None
NVHPC 24.5	Full	Full	None
oneAPI 2024.1	Full	Partial	None
ROCm 6.1	Full	Full	Partial

- Full: Required parts of the interface are implemented, but may have runtime bugs.
- Partial: Missing features for full support (e.g. multiple accelerators).
- None: No or insufficient support.
- Accelerator events require all three event types.

OpenMP Tools Interface (OMPT)

- Portable interface for analysis tools. Supported by performance tools like Score-P, HPCToolkit, TAU and Nsight Systems, and correctness tools like LLVM Archer.
- Tools register OpenMP runtime callbacks, which get invoked on directives and runtime function calls.
- Support for heterogeneous accelerators through device tracing interface, where events are written to buffers, which are flushed during program execution (similar to CUPTI).
- Event correlation via two IDs: One per target region, one for data transfers and submitted kernels.



Simplified diagram for callbacks during an OpenMP program run. Accelerator events are recorded to buffer, dispatched to the OpenMP tool via buffer complete asynchronously. With this, we can write accelerator events correlated to host callbacks.

First Profile and Trace Results – SPEC HPC 618.tealeaf_s on LUMI-G with AOMP 18.0-1 <u>File Display Plugins Help</u> Individual Processes, Accumulated Exclusive Time per Function Metric tree ■ Call tree ■ Flat tree ■ System tree ■ Statistics 50 s 100 s 150 s 200 s 250 s 1.81e8 Visits (occ) 26.83 tealeaf_base.clang_omp_target_gfx90a (1 hidden child) machine Cray generic Master thread:0 ■ 1.09 main (8 hidden children) OMP Target [0:1]:0 OpenMP Device gfx90a (0) □ 0.00 Minimum Inclusive Time (sec) → ■ 20.55 initialise comms Master thread:1 285.92 Maximum Inclusive Time (sec) → ■ 33.37 initialise application Maste...ead:1 OMP Target [1:1]:1 - 🔲 4.32 diffuse overload (31 hidden children) Master thread:2 0 bytes get (bytes) → ■ 0.27 !\$omp target @0x14d44876 (callee id=362, target type=targ) 📘 74.56 OpenMP Device gfx90a (1) ■ 74.58 OpenMP Device gfx90a (2) ☐ 0 io bytes read (bytes) - 📘 1.17 solve (4 hidden children) OMP Target [2:1]:2 → **1** 74.26 OpenMP Device gfx90a (3) - ■ 0.48 cg driver Master thread:3 Maste...ead:3 ■ 74.30 OpenMP Device gfx90a (4) 2.37e+11 bytes sent (bytes) 🕨 📘 1.06 cg init driver OMP Target [3:1]:3 → **1** 74.32 OpenMP Device gfx90a (5) 2.37e+11 bytes_received (bytes - 📘 1.10 cg_main_step_driver Master thread:4 → **1** 0.41 run_cg_calc_w ■ 74.31 OpenMP Device gfx90a (6) 1.61e9 OpenMP Memory (bytes) 74.44 OpenMP Device gfx90a (7) OMP Target [4:1]:4 5.43e+11 allocation size (bytes) - 📘 1.49 cg calc w ■ 5.43e+11 deallocation size (bytes) 🗖 1279.72 !\$omp target 🛾 0.00 MPI Rank 0 🔗 🤜 Master thread:5 \square 0 bytes leaked (bytes) 0.00 MPI Rank 1 143.52 sum_over_ranks OMP T...:1]:5 !\$omp...7556b !\$omp...74b40! OMP Target [5:1]:5 ■ 1.61e9 maximum_heap_memory_allocated (bytes) 0.00 MPI Rank 2 - 📘 0.43 run_cg_calc_ur !\$omp ...27556b !\$omp...74b40 Master thread:6 0.00 MPI Rank 3 - ■ 1.63 cg calc ur → ■ 1397.87 !\$omp target @0x0027556b (callee id=399, targe) 0.00 MPI Rank 4 **GPU Events over Time** Accumulated Exclusive Time per Function 0.00 MPI Rank 5 518.55 run_cg_calc_p ...hine Cray generic 🕨 📘 1133.44 halo update driver 0.00 MPI Rank 6 1,000 s node nid005002 0.00 MPI Rank 7 0.90 solve finished driver !\$omp target @0x0027556b node nid005003 0.81 field summary drive ...ice gfx90a (0) 1,298.834 s !\$omp target (kernel execution) @0x0027556b ■ 74.53 OpenMP Device gfx90a (0) 🕨 📘 0.41 field summary driver ...ice gfx90a (1) 1,241.426 s !\$omp target @0x00274b40 3.58 kernel finalise driver 📘 74.31 OpenMP Device gfx90a (1) ...ice gfx90a (2) 1,191.076 s !\$omp target (kernel execution) @0x00274b40 🕨 📘 1.12 finalise comms ■ 74.41 OpenMP Device gfx90a (2) ...ice gfx90a (3) 🗖 74.48 OpenMP Device gfx90a (3) 516.456 s !\$omp target @0x00275fd2 0.61 KERNELS (6 hidden children ...ice gfx90a (4) 74.32 OpenMP Device gfx90a (4) somp target submit @0x00274b40 · ■ 74.54 OpenMP Device gfx90a (5) → ■ 14.00 !\$omp target (kernel execution) @0x00281001 ...ice gfx90a (5) **1** 74.56 OpenMP Device gfx90a (6) → 13.42 !\$omp target (kernel execution) @0x0027fd03 ▶ ■ 6.00 !\$omp target (kernel execution) @0x0027d545 0.00 MPI Rank 8 node nid005003 somp target submit @0x00274b4 0.00 MPI Rank 9 0.00 MPI Rank 10 0.00 MPI Rank 11 0.00 MPI Rank 12 0.00 MPI Rank 13 0.00 MPI Rank 14 0.27 !\$omp target (kernel execution) @0x0028b0e7 (caller id=308, 7673.64 (100.00%) 7673.64 0.00 7673.64 0.00 74.87 (6.29%) 1191.08 Ready

References & Acknowledgements

- Knüpfer, Andreas, et al. "Score-P: A joint performance measurement run-time infrastructure for Periscope, Scalasca, TAU, and Vampir." Tools for High Performance Computing 2011
- Feld, Christian, et al. "Score-P and OMPT: Navigating the Perils of Callback-Driven Parallel Runtime Introspection" International Workshop on OpenMP 2019
- OpenMP Application Programming Interface, Version 5.2: https://www.openmp.org/wp-content/uploads/OpenMP-API-Specification-5-2.pdf
- This poster is based upon work supported by the BMBF as part of the ENSIMA project (Grant No. 16ME0630)
- We acknowledge the EuroHPC Joint Undertaking for awarding this project access to the EuroHPC supercomputer LUMI, hosted by CSC (Finland) and the LUMI consortium through a EuroHPC Regular Access call.