OpenMP Tools based on OMPT

Joachim Protze
OpenMP Booth, SC‘19

i12 High Performance Computing | RWTH AACHEN UNIVERSITY

Implementation state of OMPT

- LLVM: since 8.0
- icc: since 19.0.1

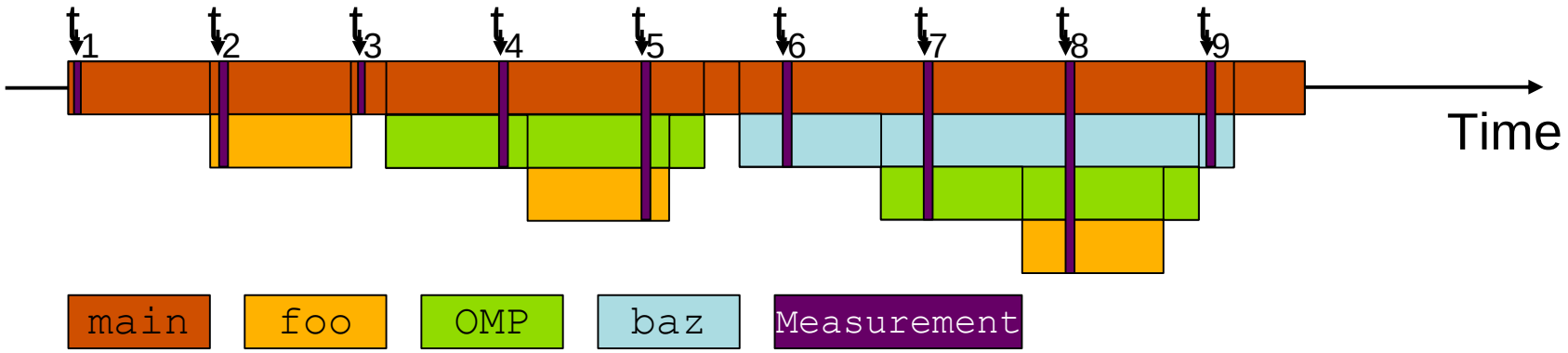
- OMPT support for target construct currently missing

OMPT support for sampling

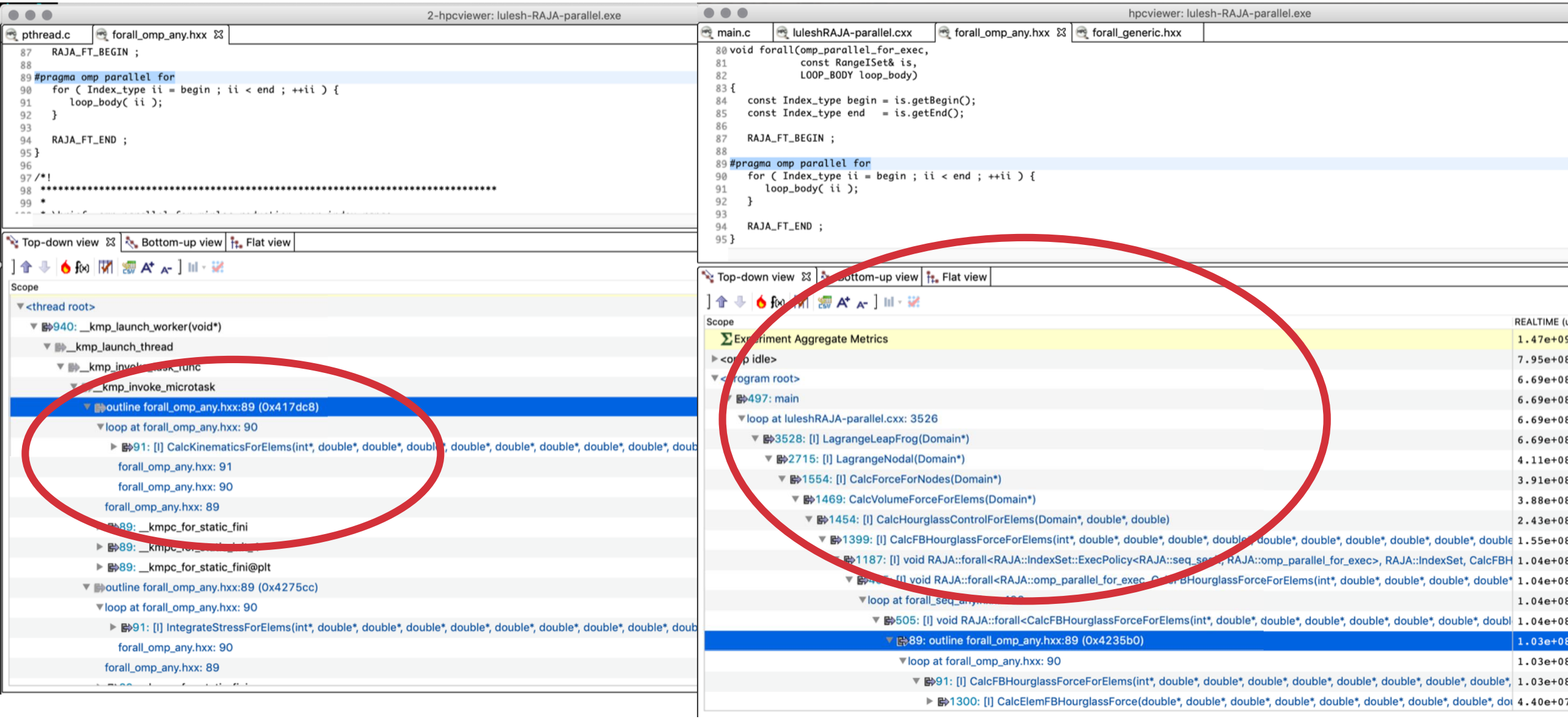
- OMPT defines states like *barrier-wait*, *work-serial* or *work-parallel*
  - Allows to collect OMPT state statistics in the profile
  - Profile break down for different OMPT states

- OMPT provides frame information
  - Allows to identify OpenMP runtime frames.
  - Runtime frames can be eliminated from call trees

```
void foo() {}
void baz() {#omp foo();}
int main()
{foo(); #omp foo();
baz(); return 0;}
```

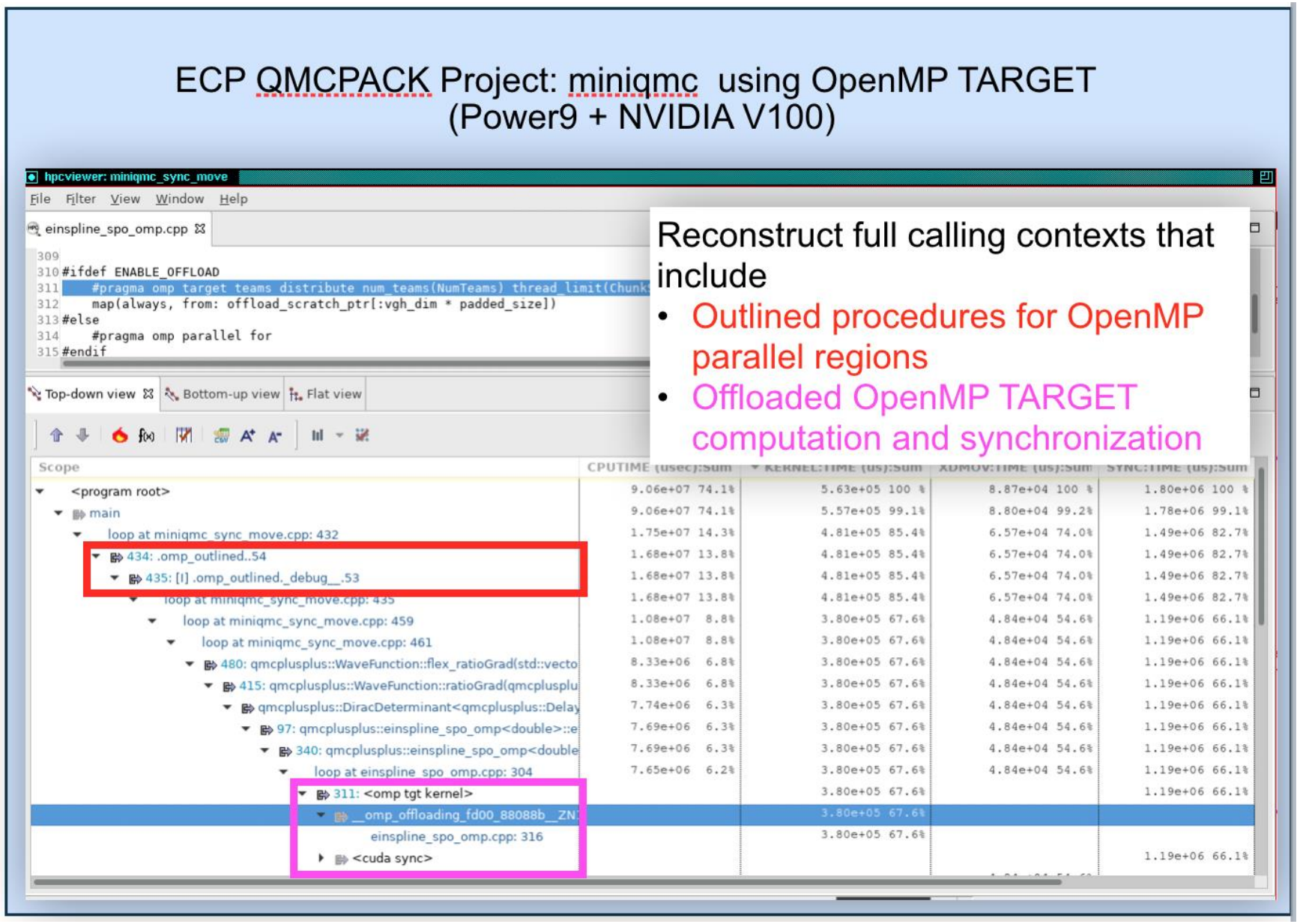


HPCToolkit: OpenMP-aware callstack with OMPT



HPCToolkit's Support for OpenMP TARGET

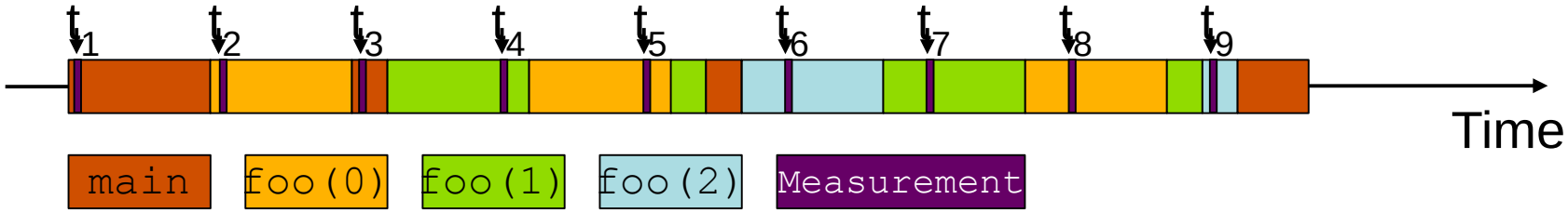
- HPCToolkit implementation of OMPT OpenMP API
  - host monitoring
    - leverages callbacks for regions, threads, tasks
    - employs OMPT API for call stack introspection
  - GPU monitoring
    - leverages callbacks for device initialization, kernel launch, data operations
  - reconstruction of user-level calling contexts
- Leverages implementation of OMPT in LLVM OpenMP and libomptarget



Sampling vs. Instrumentation

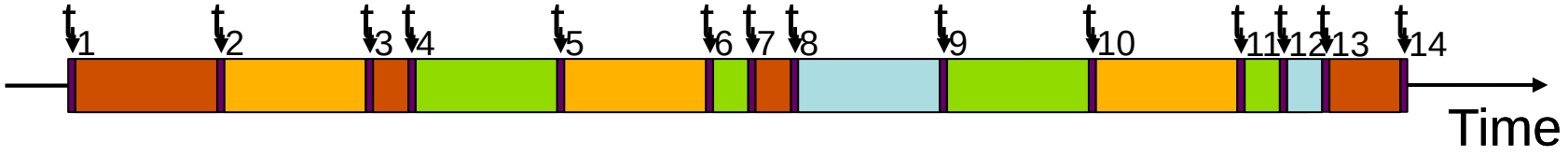
Sampling

- Running program is periodically interrupted to take measurement
- *Statistical* inference of program behavior
- Works with unmodified executables



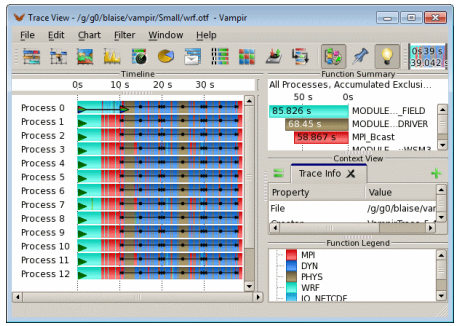
Instrumentation

- Every event of interest is captured directly
- More detailed and *exact* information
- Typically: recompile for instrumentation

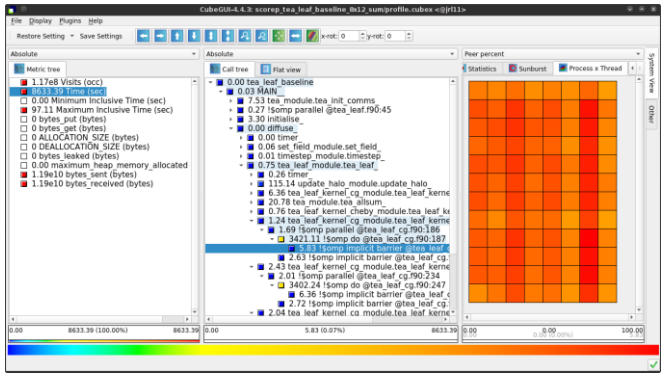


Score-P

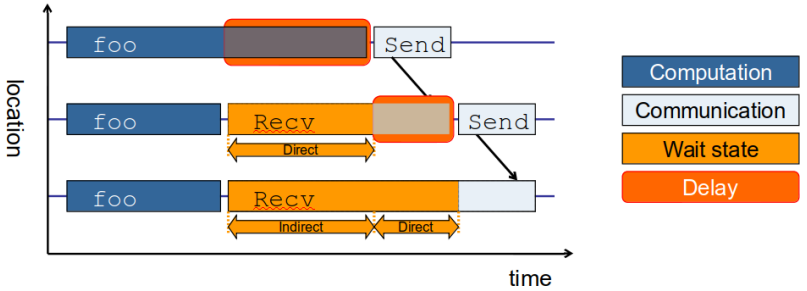
- Scalable Performance Measurement Infrastructure for Parallel Codes
  - Instrumenter, libraries, and tools to generate profile and trace measurements
- Provides measurement data for:



VAMPIR



cube scalasca

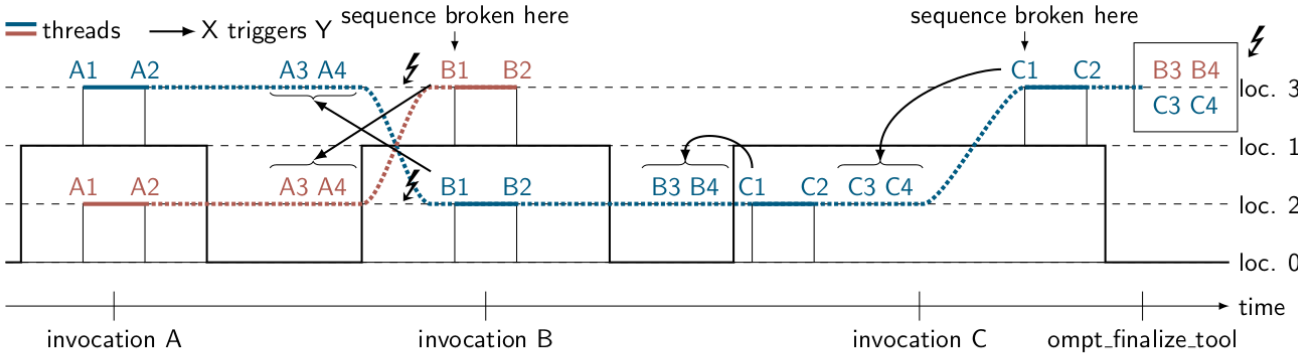


trace tools scalasca

- Available under 3-clause BSD open-source license
- Documentation & sources: http://www.score-p.org
- Contact: support@score-p.org

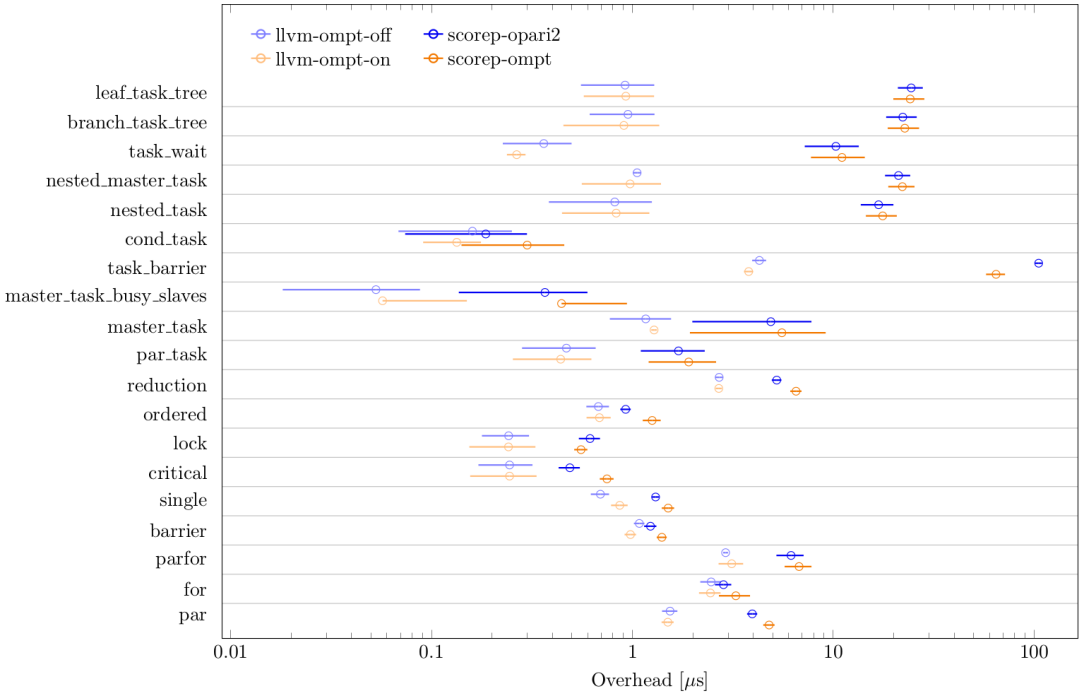
Score-P and OMPT

- Implementation Challenge: Overdue events and non-deterministic scheduling, logical vs. system view, otherwise straight forward
- Similar performance compared to OPARI2
  - EPCC OpenMP micro-benchmark suite
  - SPEC OMP2012
- **OMPT** can serve as **drop-in replacement** for OPARI2 (work in progress)



Expected event order per invocation on location 2 (loc. 2) and location 3 (loc. 3)

1 implicit-task-begin **2** implicit-barrier-begin **3** implicit-barrier-end **4** implicit-task-end



Overhead reported by the EPCC OpenMP Benchmark Suite, 12 threads

IWOMP19 paper: *Score-P and OMPT: Navigating the perils of callback-driven parallel runtime introspection* https://doi.org/10.1007/978-3-030-28596-8_2

TAU v2.29: OpenMP Support Update

- Built-in support for OMPT 5.0 specification

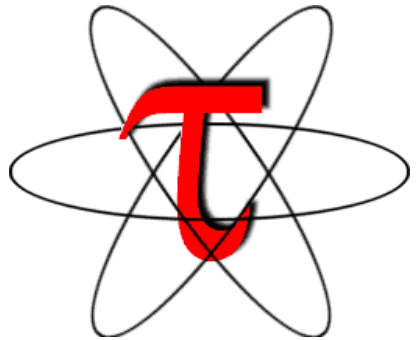
```
$ ./configure -bfd=download -openmp -ompt ...
```

- Detects if compiler supports OMPT / OpenMP 5.0
- Compiles support for latest LLVM OpenMP runtime otherwise
  - Only for GCC, LLVM/CLANG, Intel compilers without support
- Developing target support
- Linker-based instrumentation (static executables)

```
$ tau_cc.sh -o foo *.o libstuff.a
```

- Runtime preloading (dynamic executables)

```
$ tau_exec -T openmp,ompt -ompt ./my_program
```



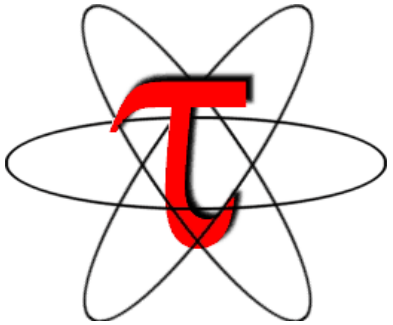
TAU Profile Views

pprof output:

instrumentfunctions — khuck@cyclops:~/src/tau2/examples/mm — ssh • ssh cyclops — 130×38

```
NODE 0;CONTEXT 0;THREAD 0:
---------------------------------------------------------------------------------------
%Time    Exclusive    Inclusive       #Call      #Subrs  Inclusive Name
              msec   total msec                          usec/call
---------------------------------------------------------------------------------------
100.0           8        1,938           1           1    1938013 .TAU application
 99.6       0.063        1,929           1           1    1929719 main [{/home/users/khuck/src/tau2/examples/mm/matmult.c} {225,0
 99.6       0.247        1,929           1          11    1929656 do_work [{/home/users/khuck/src/tau2/examples/mm/matmult.c} {13
 95.4           2        1,848           5           5     369650 OpenMP_Implicit_Task
 52.7       0.108        1,021           1           1    1021354 compute [{/home/users/khuck/src/tau2/examples/mm/matmult.c} {10
 52.7       0.006        1,021           1           1    1021246 OpenMP_Parallel_Region compute [{/home/users/khuck/src/tau2/exa
 52.7       0.032        1,020           1           1    1020801 .omp_outlined. [{/home/users/khuck/src/tau2/examples/mm/matmult
 52.7       1,020        1,020           1           0    1020769 .omp_outlined._debug__ [{/home/users/khuck/src/tau2/examples/mm
 42.5       0.039          823           1           1     823555 compute_interchange [{/home/users/khuck/src/tau2/examples/mm/ma
 42.5       0.005          823           1           1     823516 OpenMP_Parallel_Region compute_interchange [{/home/users/khuck/
 42.5       0.037          822           1           1     822810 .omp_outlined..2 [{/home/users/khuck/src/tau2/examples/mm/matmu
 42.5         822          822           1           0     822773 .omp_outlined._debug__.1 [{/home/users
  4.2          78           81           3           3      27314 initialize [{/home/users/khuck/src/ta
  0.2       0.399            3           3           3       1300 OpenMP_Parallel_Region initialize [{/
  0.1       0.029            2           3           3        878 .omp_outlined. [{/home/users/khuck/sr
  0.1           2            2           3           0        869 .omp_outlined._debug__ [{/home/users/
  0.1           2            2           3           0        742 allocateMatrix [{/home/users/khuck/sr
  0.0       0.331        0.331           3           0        110 freeMatrix [{/home/users/khuck/ta

NODE 0;CONTEXT 0;THREAD 1:
---------------------------------------------------------------------------------------
%Time    Exclusive    Inclusive       #Call      #Subrs  Inclusive Name
              msec   total msec                          usec/call
---------------------------------------------------------------------------------------
100.0       0.083        1,862           1           1    1862363 .TAU application
100.0       0.084        1,862           1           5    1862280 OpenMP_Thread_Type_ompt_thread_worker
100.0         117        1,862           5           5     372439 OpenMP_Implicit_Task
 51.2       0.029          953           1           1     953762 .omp_outlined. [{/home/users/khuck/sr
 51.2         953          953           1           0     953733 .omp_outlined._debug__ [{/home/users/
 42.4       0.035          788           1           1     788843 .omp_outlined..2 [{/home/users/khuck/
 42.4         788          788           1           0     788808 .omp_outlined._debug__.1 [{/home/user
  0.1       0.002            2           3           3        829 .omp_outlined. [{/home/users/khuck/sr
:
```



Thread 0 callpath:

TAU: ParaProf: Statistics for: node 0, thread 0 - /Users/khuck/Google Drive/SC19/openmp

Name	Exclusive TIME	Inclusive TIME ▽	Calls	Child ...
.TAU application	0.008	1.898	1	1
main	0	1.89	1	1
do_work	0	1.89	1	11
compute	0	0.99	1	1
OpenMP_Parallel_Region compute	0	0.99	1	1
OpenMP_Implicit_Task	0	0.99	1	1
.omp_outlined.	0	0.989	1	1
.omp_outlined._debug__	0.989	0.989	1	0
compute_interchange	0	0.813	1	1
OpenMP_Parallel_Region compute_interchange	0	0.813	1	1
OpenMP_Implicit_Task	0.015	0.813	1	1
.omp_outlined..2	0	0.798	1	1
.omp_outlined._debug__.1	0.797	0.797	1	0
initialize	0.081	0.084	3	3
OpenMP_Parallel_Region initialize	0	0.004	3	3
OpenMP_Implicit_Task	0.001	0.003	3	3
.omp_outlined.	0	0.003	3	3
.omp_outlined._debug__	0.003	0.003	3	0
allocateMatrix	0.002	0.002	3	0
freeMatrix	0	0	3	0

Data race detection tool: Archer

- Error checking tool for
  - Memory errors
  - **Threading errors** (OpenMP, Pthreads)
- Based on ThreadSanitizer (runtime check)
- Available for Linux, Windows and Mac
- Supports C, C++
- Workflow available for Fortran
- Synchronization information based on **OMPT**

- More info: https://github.com/PRUNERS/archer

- Will be part of the next release of LLVM: 9.1 / 10.0
  - Accepted and committed to LLVM master branch

More OpenMP Correctness Tools

- ROMP: Yizi Gu, John M. Mellor-Crummey:
  Dynamic data race detection for OpenMP programs. SC 2018

- TaskSanitizer: Hassan Salehe Matar, Didem Unat:
  Runtime Determinacy Race Detection for OpenMP Tasks. Euro-Par 2018

- Ricardo Bispo Vieira, Antoine Capra, Patrick Carribault, Julien Jaeger, Marc Pérache, Adrien Roussel:
  Detecting Non-sibling Dependencies in OpenMP Task-Based Applications. IWOMP 2019

Tools for OpenMP Tasking

- Vishakha Agrawal, Michael J. Voss, Pablo Reble, Vasanth Tovinkere, Jeff Hammond and Michael Klemm
  Visualization of OpenMP Task Dependencies using Intel Advisor Flow Graph Analyzer, IWOMP 2018

- Peder Voldnes Langdal, Magnus Jahre and Ananya Muddukrishna:
  Extending OMPT to support Grain Graphs, IWOMP 2017