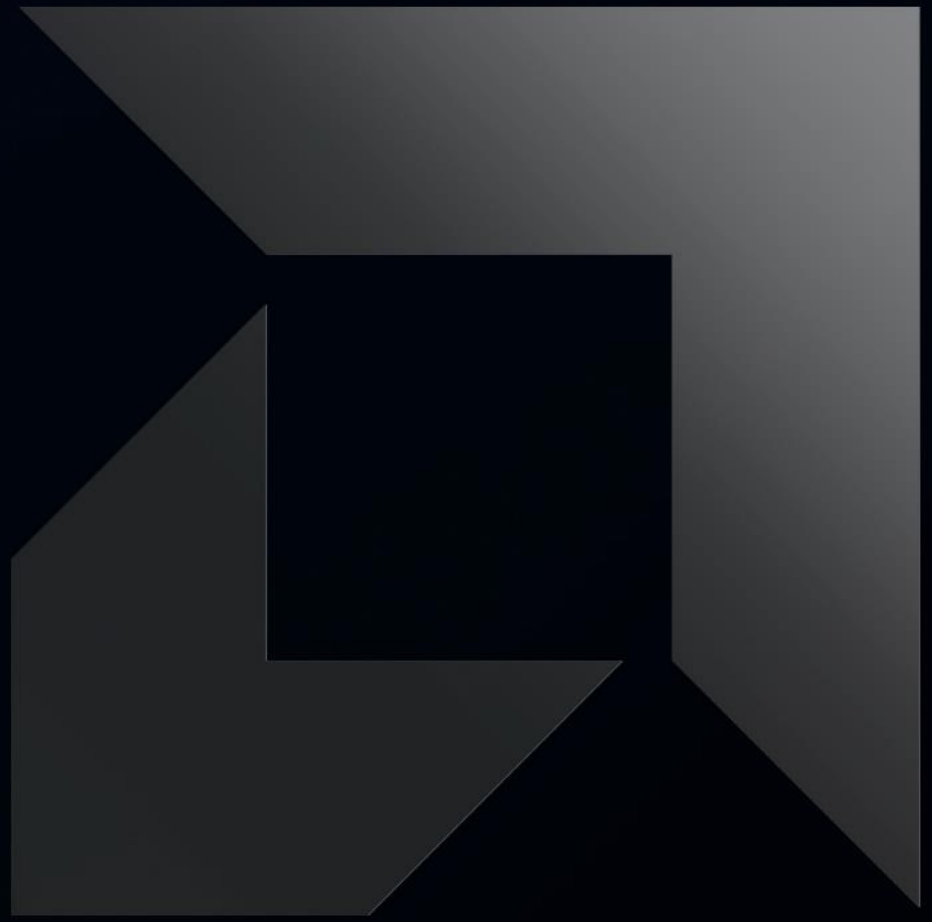


OMPT Support in ROCm™

Past, Present & Future

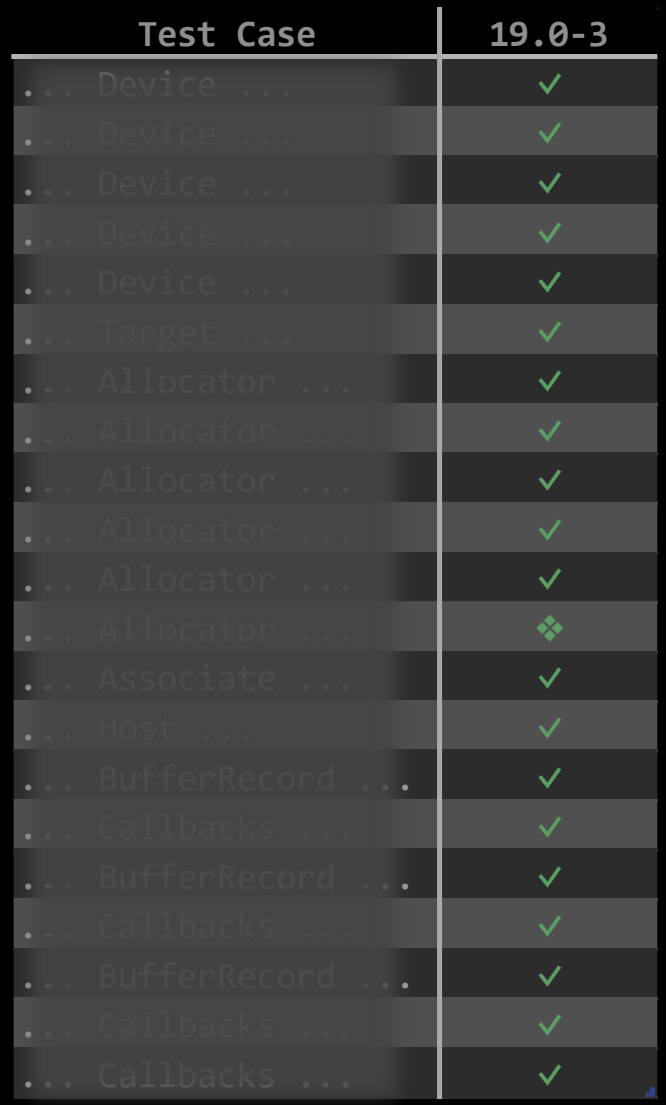
ROCm OpenMP© GPU Compiler Team

Michael Halkenhäuser, Dhruva Chakrabarti,
Jan-Patrick Lehr, Ron Lieberman

AMD
together we advance_

„OMPT is important to us“

- **Our Goals:**
  - **Improve OMPT** { **feature support**, **test coverage**
- **Ensure a solid foundation for tool developers**



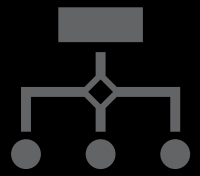
Introduction

ROCm

AMD ROCm is an open software platform for GPU compute consisting of compilers, libraries, tools, etc.

(https://www.amd.com/en/products/software/rocm.html)

AOMP

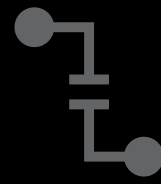


AOMP is an open-source Clang/LLVM-based compiler with added support for the OpenMP® API on Radeon™ GPUs that builds on top of ROCm

Releases are more frequent than **ROC**m

(https://github.com/ROCm/aomp)

LLVM



Since AOMP is Clang/LLVM based, the high-level software architecture of OpenMP target offload support is identical

AOMP is both ahead and behind LLVM mainline trunk (upstream)

“downstream”

“upstream”

OMPT Events 101

```
int main(int argc, char **argv) {
  int Values[1024] = {0};

  #pragma omp target loop
  for(int i = 0; i < 1024; ++i) {
    Values[i] = 1;
  }

  return 0;
}
```

CALLBACKS
- synchronous
- ordered

TRACE RECORDS
- asynchronous
- unordered

Fun fact: Code snippet generates more than 30 device events

OMPT Device Events 101

CALLBACKS
- synchronous
- ordered

TRACE RECORDS
- asynchronous
- unordered

```
Callback Kernel Submit EMI
  endpoint=1
  req_num_teams=1
  …
```

```
rec=0x1257a00
  type=10 (Target kernel)
  time=821383491484741O
  …
```

EMI Callbacks
- **Provide event information directly**
- **Become default in OpenMP 6.0**

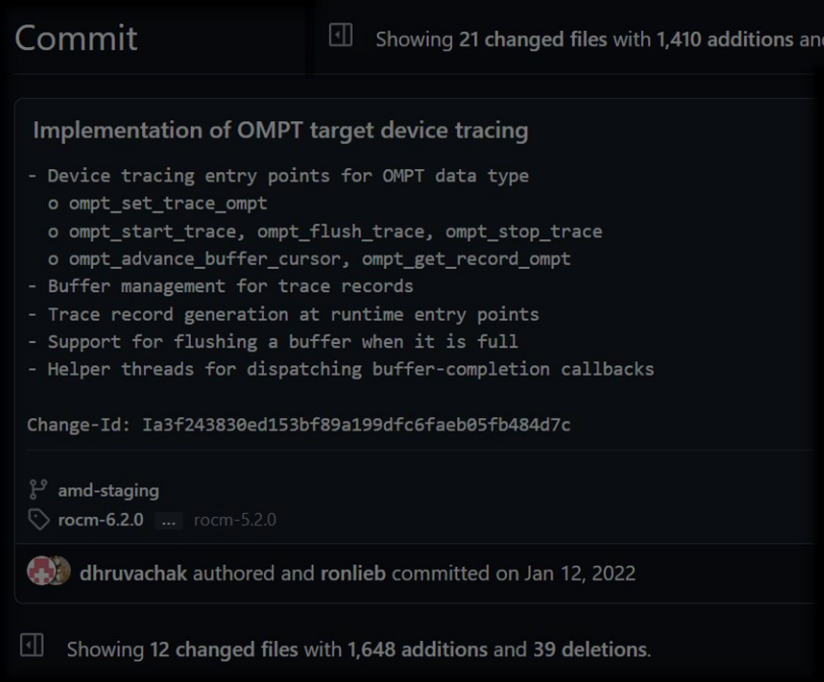
Trace Records
- **Provide info on extended events**
  - **Begin- & End-time**
  - **Misc. data structures**
  - **…**

Tools using OMPT events, e.g.
- **score-p, TAU and HPCToolkit**

Beginning of Downstream OMPT Support

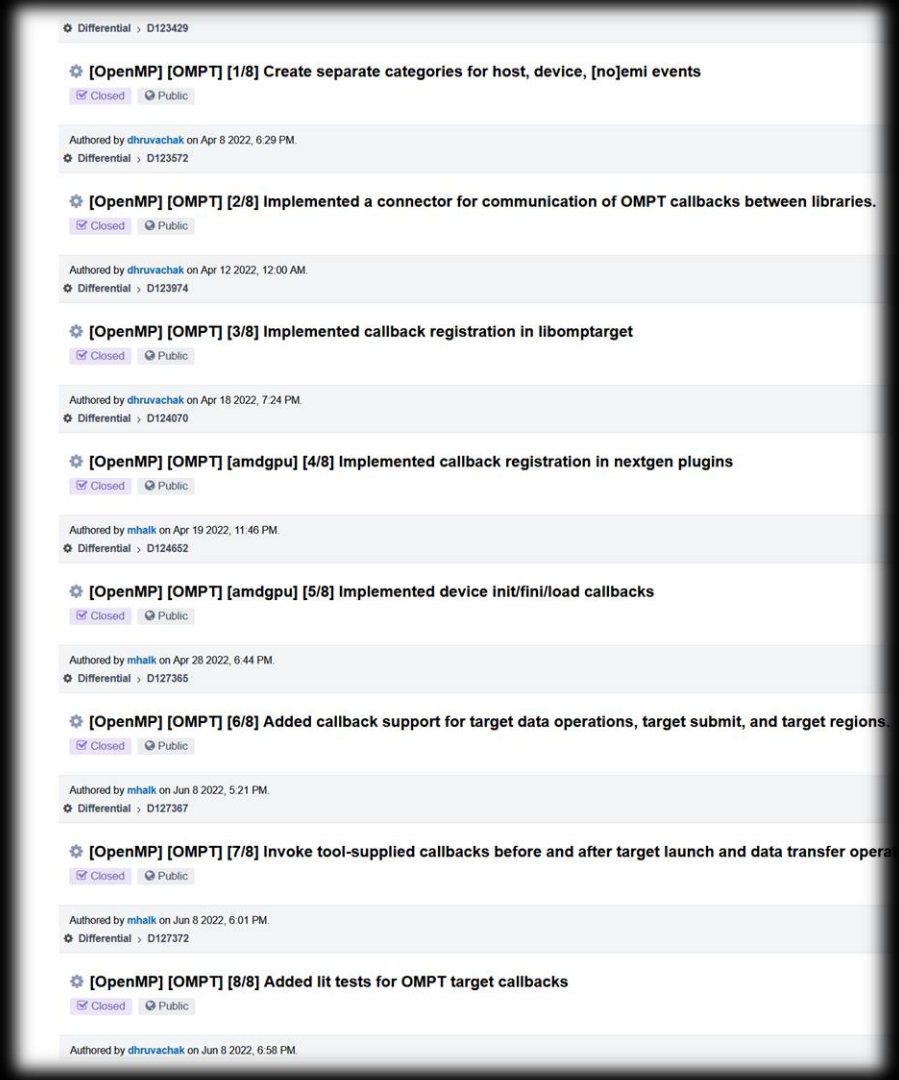
- **Callbacks implemented in HPCToolkit's LLVM fork (by John Mellor-Crummey)**
- **Callbacks refactored for downstream**
- **Trace Records implemented in downstream**
- **Commitment to open-source software**

  ⟶ **Upstream our changes**



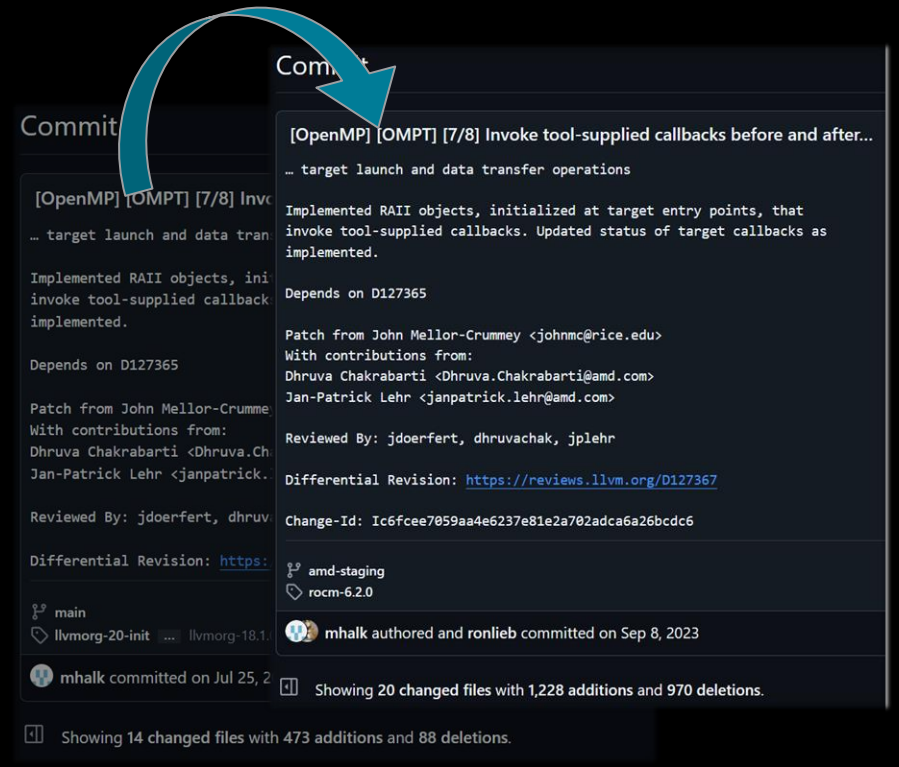
Upstreaming of OMPT Device Callback Support

- Segmentation of downstream-diff
  - Result: 8 Phabricator patches
- Upstream is a moving target
  - Addition of ‘nextgen-plugins’ needed refactoring of patches 4-7
- Refactoring
  - Templated design desired (by Johannes Doerfert)
- Reordering
  - “Test first, add callbacks later”

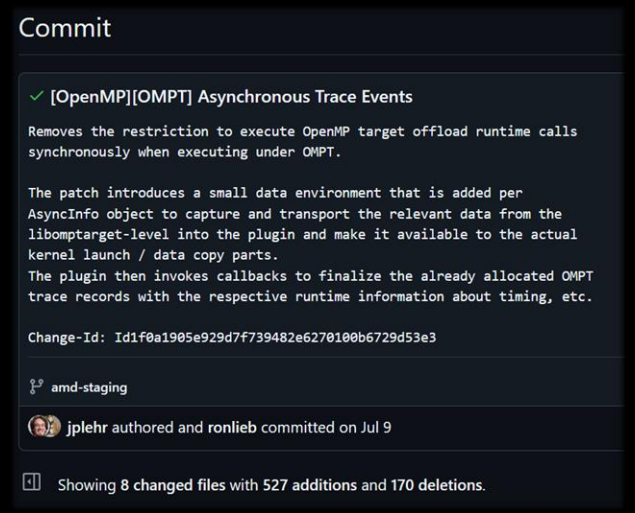


Meanwhile “behind the scenes”

- Adoption of upstream callback support
  - Refactor trace record handling
  - Prepare downstream set of OMPT patches



- Expansion of OMPT capabilities
  - Implement asynchronous trace records



Downstream OMPT Support History

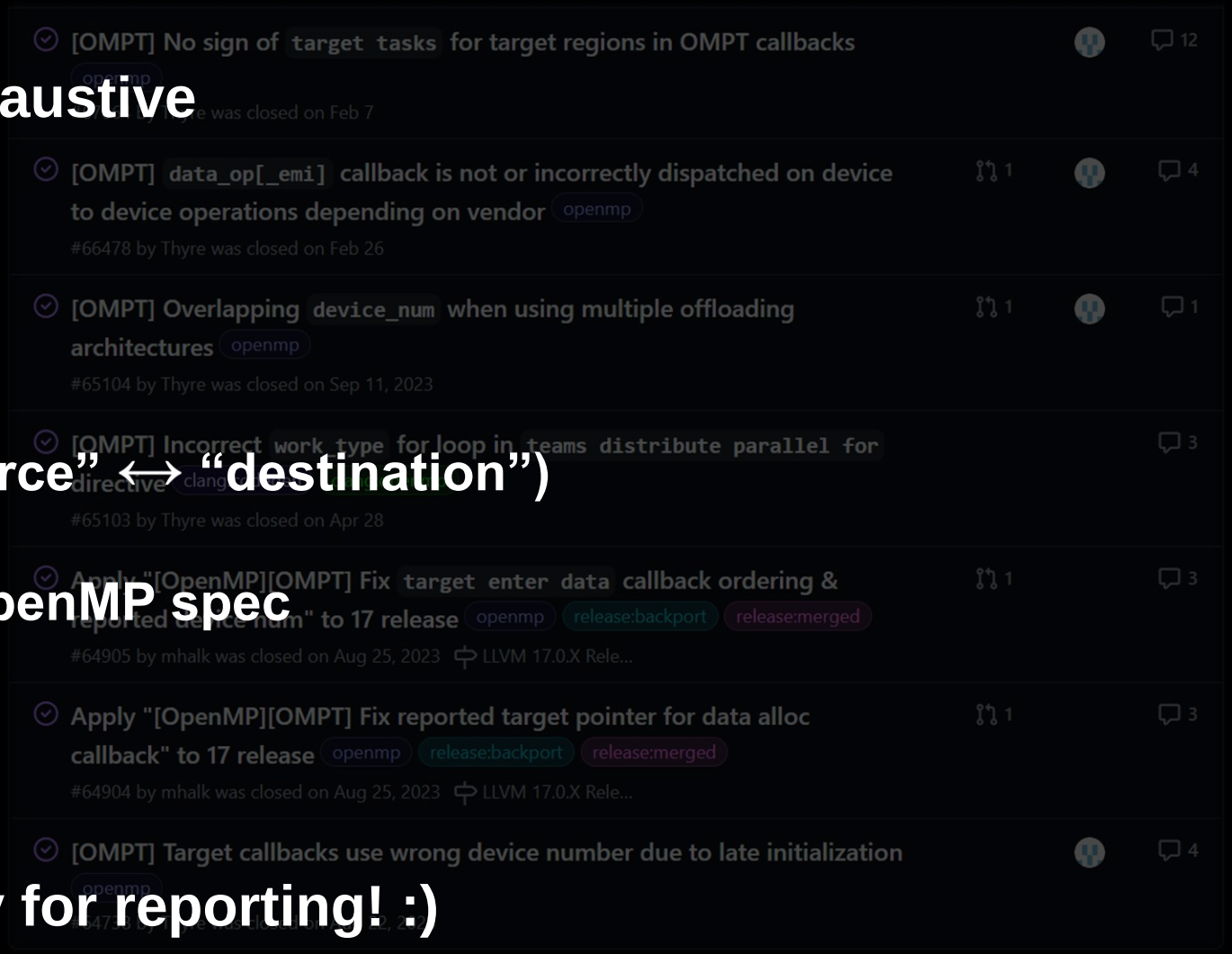
Test Case	18.0-1	19.0-0	19.0-2	19.0-3
OnDeviceDefaultTeams_Sequenced	✓	✓	✓	✓
OnDeviceSetNumTeams_Sequenced	✓	✓	✓	✓
OnDevice_Sequenced_Grouped	✓	✓	✓	✓
ExplicitDefaultDevice	✓	✓	✓	✓
Multi…es	✓	✓	✓	✓
DeclareTargetGlobalAndUpdate	✓	✓	✓	✓
ExplicitAllocatorAPI	✓	✓	✓	✓
ExplicitAllocatorAndCopyAPI	✗	✓	✓	✓
Expli…AndUpdate_NoDataOp	✗	✗	✗	✓
ExplicitAllocatorAndUpdate_DataOp	✗	✗	✗	✓
ExplicitAllo…AndUpdate_DataOpStack	✗	✗	✗	✓
Explicit…ce	❖	❖	❖	❖
ExplicitAssociateDisassociatePtrAPI	✗	✗	✗	✓
OnHost_On…	✓	✓	✓	✓
OnDeviceBufferRecord_parallel_for	✓	✓	✓	✓
OnDeviceCallbacks_parallel_for	✓	✓	✓	✓
OnDeviceBufferRecord_teams_distribute_parallel_for	✓	✓	✓	✓
OnDeviceCallbacks_teams_distribute_parallel_for	✓	✓	✓	✓
OnDeviceBufferRecord_teams_distribute_parallel_for_nowait	✓	✓	✓	✓
OnDeviceCallbacks_teams_distribute_parallel_for_nowait	✓	✓	✓	✓
DeviceLoad	✓	✓	✓	✓

- Host
- Device
  - API
  - Callbacks
  - Trace Records

Testing & Regressions

- Test coverage almost never exhaustive
  - Regressions happen(ed)

- Regression examples
  - Missing EMI callback endpoints
  - Switched device numbers (“source” ↔ “destination”)
  - RAII object lifetimes
  - “Wrong” interpretation of the OpenMP spec
  - …



- Big *thank you* to the community for reporting! :)

OMPT Testing

- **Test scenarios rarely straightforward**
  - **LLVM lit & FileCheck may prove cumbersome**

- **Key features**
  - **Maintainability**

    ”For each desired change, make the change easy (warning: this may be hard), then make the easy change.“

    -- Kent Beck

  - **Robustness**

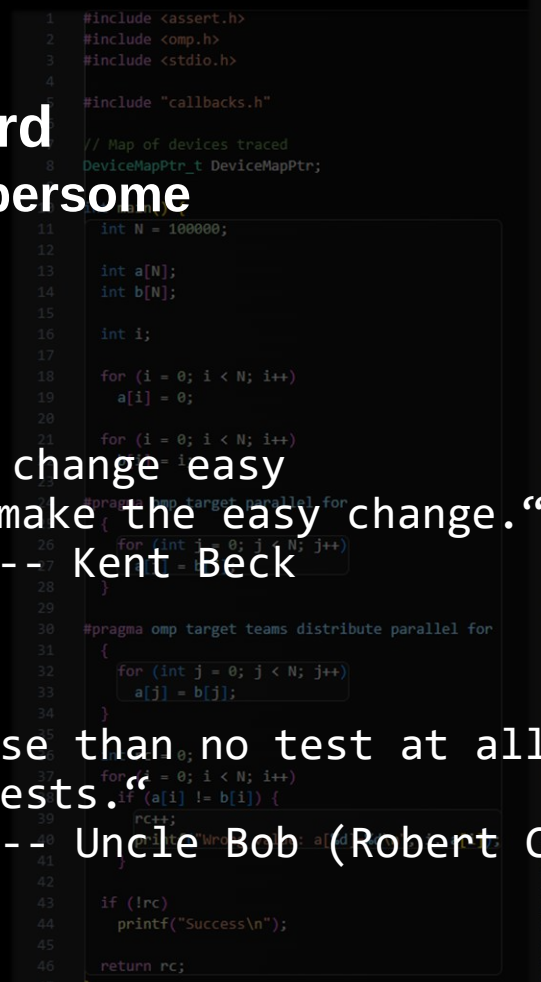
    ”A test that fails randomly is worse than no test at all. It teaches you to distrust your tests.“

    -- Uncle Bob (Robert C. Martin)

  - **Speed**

    ”The faster you can run your tests, the more tests you will run.“

    -- Kent Beck



ompTest

- Goal: OMPT unit testing library
  “GoogleTest-like”

- Own set of advantages
  - Unified event creation & verification
  - Improved readability & ease-of-use
  - Customized equality operators

- Limitations due to library-nature
  - OpenMP RT init / fini: out of scope

```
// Check for callback
OMPT_ASSERT_SET(TargetSubmit)

// Check for trace record
OMPT_ASSERT_SET(BufferRecord,
                /*Type=*/CB_KERNEL)
```

```
#pragma omp target loop
for(int i = 0; i < N; ++i) {
  Values[i] = 1;
}
```

ompTest

- **Related papers for SC'24**
  - **Correctness Workshop**
    https://correctness-workshop.github.io/2024

    "ompTest – Unit Testing with OMPT"

  - **HPC Bugs Fest**
    https://sites.google.com/view/hpc-bugs-fest/home

    "OMPTBench – OpenMP Tool Interface Conformance Testing"

```
// Check for callback
OMPT_ASSERT_SET(TargetSubmit)

// Check for trace record
OMPT_ASSERT_SET(BufferRecord,
                /*Type=*/CB_KERNEL)
```

```
#pragma omp target loop
for(int i = 0; i < N; ++i) {
  Values[i] = 1;
}
```

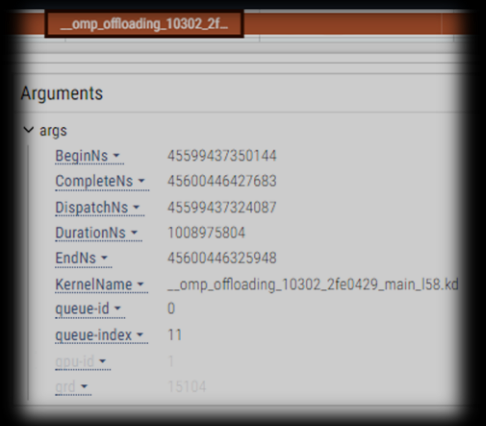
Possible Future of OMPT

- Device-side tracing support (upstream)
- “ompTest” library (upstream)
- OMPT “native” data type

```
libomptarget_ompt_set_trace_ompt(/*DeviceId=*/i,
                                 /*Enable=*/1,
                                 /*EventyTy=*/0

OMPT_ASSERT_SET_GROUPED("R1", TargetEmi, /*Kind=*/TARGET,
                        /*Endpoint=*/BEGIN)

OMPT_ASSERT_SET_GROUPED("R1", BufferRecord, /*Type=*/CB_TARGET,
                        /*Kind=*/TARGET,
                        /*Endpoint=*/BEGIN)
OMPT_ASSERT_SET_GROUPED("R1", BufferRecord, /*Type=*/CB_DATAOP,
                        /*OpType=*/ALLOC, /*Bytes=*/sizeof(a),
                        /*MinimumTimeDelta=*/10)
OMPT_ASSERT_SET_GROUPED("R1", BufferRecord, /*Type=*/CB_DATAOP,
                        /*OpType=*/H2D, /*Bytes=*/sizeof(a),
                        /*Timeframe=*/{10, 100000})
OMPT_ASSERT_SET_GROUPED("R1", BufferRecord, /*Type=*/CB_DATAOP,
                        /*OpType=*/ALLOC, /*Bytes=*/sizeof(b),
                        /*MinimumTimeDelta=*/10)
OMPT_ASSERT_SET_GROUPED("R1", BufferRecord, /*Type=*/CB_DATAOP,
                        /*OpType=*/H2D, /*Bytes=*/sizeof(b),
                        /*Timeframe=*/{10, 100000})
```



Disclaimer

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED ‘AS IS.” AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.


AMD