

OpenMP[®]

SC'20 Booth Talk Series

Recent, Current and Future
OpenMP Directions:
OpenMP 5.1 and More!

Bronis R. de Supinski
Chair, OpenMP Language Committee



Recent, Current and Future OpenMP Directions: OpenMP 5.1 and More!

Bronis R. de Supinski
Chair
OpenMP Language Committee

November 2020



OpenMP 5.1 ratified in November 2020

- Proceedings of the IEEE article on vision: “The Ongoing Evolution of OpenMP”
 - Broadly support on-node performant, portable parallelism
 - OpenMP 5.0 fits within that vision
 - OpenMP 6.0 will be a major step to further realizing it
- OpenMP 5.1 refines how OpenMP 5.0 realizes the vision
 - Does not break (most?) existing code
 - Several keywords and symbols have been deprecated
 - Many clarifications and corrections, some significant enhancements
 - Built on two comment drafts (TR8 and TR9)
 - 251 (?) GitHub issues; All issues after TR9 arose from final quality control pass

OpenMP 5.1 adds some significant extensions

■ The `interop` construct

- Improves native device support (e.g., CUDA streams)
- Also supports interoperability with CPU-based libraries (e.g., TBB)
- Deep-dive talk by Tom Scogland from LLNL

■ The new `dispatch` construct, improved `declare variant` directive

- Enable use of variants with device-specific arguments
- Elision of “unrecognized” code
- Deep-dive talk by Ravi Narayanaswamy from Intel

OpenMP 5.1 adds some significant extensions

■ The `assume` directive

- Supports optimization hints based on invariants
- Supports `promise` to limit OpenMP usage to (optimizable) subsets
- Deep-dive talk by Johannes Doerfert from Argonne National Laboratory

■ Loop transformation directives: The `tile` and `unroll` directives

- Control use of traditional sequential optimizations
- Ensure that they are applied when, where appropriate relative to parallelization
- Deep-dive talk by Michael Kruse from Argonne National Laboratory

OpenMP 5.1 adds several other extensions

- Adds full support for C11, C++11, C++14, C++17, C++20 and Fortran 2008 and partial support for Fortran 2018
- Extends directive syntax to C++ attribute specifiers
- The `error` directive supports user-defined warnings and errors
- The `scope` construct supports reductions within parallel regions
- Support for mapping (translated) function pointers
- Device-specific environment variables to control their ICVs
- The `nothing` directive supports `metadirective` clarity and completeness

OpenMP 5.1 refines existing functionality

- The `masked` construct supports filtering execution per thread
 - Replaces deprecated `master` construct
- Extends `atomic` construct to support compare-and-swap, min and max
- Adds many clauses and clause modifiers including `nowait` to `taskwait` construct, `strict` to `taskloop` construct clauses, iterators to several clauses
- Adds several new runtime routines, including more memory allocation flavors
- Deprecations include:
 - The `master` affinity policy and `master` construct
 - Cray pointers

OpenMP 5.2 will be released by November 2021

- Late decision during 5.1 process to add this additional minor release
- Will focus on improving specification of OpenMP syntax
 - Consolidate syntax to highlight commonality and to facilitate use of attributes
 - Clarify and simplify specification of restrictions on clause usage
- Other changes likely to reduce redundancy in specification
- May try to reduce caveats to Fortran 2018 support
- Should view as an extended quality control pass
- 4 issues already created, likely to involve 30 to 50 total

OpenMP 6.0 will be released in November 2023

- Deeper support for descriptive and prescriptive control
- More support for memory affinity and complex hierarchies
- Support for pipelining, other computation/data associations
- Continued improvements to device support
 - Extensions of deep copy support (serialize/deserialize functions)
- Task-only or free-agent threads
- Event-driven parallelism
- Removal of features that were deprecated in 5.0 or 5.1
- 78 issues already deferred to 6.0



**Lawrence Livermore
National Laboratory**

The OpenMP logo features the word "Open" in a white, lowercase, sans-serif font, followed by "MP" in a larger, bold, uppercase, sans-serif font. A horizontal white line underlines the "Open" portion. A registered trademark symbol (®) is located to the right of the "MP".

OpenMP[®]

SC'20 Booth Talk Series

For the OpenMP specification, tutorials, forum, reference guides, and links to other resources, visit **www.openmp.org**