OpenMP 6.0 Progress and Directions

Bronis R. de Supinski
Chair
OpenMP Language Committee
OpenMP 6.0 will be released in November 2024

- OpenMP decided to release OpenMP 6.0 one year later than previous target
- Several factors have contributed to the decision
  - ARB essentially agreed to wait until enough major new features will have been adopted
  - Progress to date does not support release of final comment draft next summer
  - Work on OpenMP 5.2 impacted available effort for 6.0 activities in 2021
  - The above reduction was in addition to the difficulties related to the pandemic
  - Exacerbated by reduced in-person participation at two 2022 hybrid face-to-face meetings
- Confident new target will not slip further
  - TR11 shows similar progress to first TR for previous major versions
  - Major new feature targets have been clearly identified and are on track for 2024
  - TR12 will be released in November 2023 and final comment draft in summer of 2024
Major new features will characterize OpenMP 6.0

- Free-agent tasks (or threads, depending on the weather)
  - Support for top-level task parallelism (i.e., explicit `parallel` directive not needed)
  - “Any” thread can execute explicit tasks for which `free_agent` clause evaluates to true
  - Restricts use of some runtime routines (or extends possible return values)

- Major improvements for use of a single device
  - Explicit progress guarantee adopted in TR11, will likely extend with related directives
  - Default device and visible devices to simplify control of device use and availability
  - Mechanisms to simplify use of device memory (by providing greater certainty or clarity)
    - New `groupprivate` directive in TR11 is an initial mechanism in this direction
    - Unified host and device allocators and added significant cross-device improvements
  - Extensions of deep copy support (serialize/deserialize functions)
  - Add `coexecute` directive (i.e., descriptive array language offload support)
OpenMP 6.0 will also include some of these major new features

- Support for event-based parallelism
  - Need to support a team that executes an event loop
  - Event loop generates tasks that may be computationally expensive
  - Fast response time in event loop is essential
  - Likely to involve generating tasks for other teams

- True support for using multiple devices
  - Scoping/cross-device support for atomic and other memory operations
  - Support for bulk launch
  - Support for updating data across multiple devices (broadcast/multicast, other collectives)
  - Support for work distribution across devices
  - Considering relaxing restrictions on nested target regions
OpenMP 6.0 will also include other significant new features

- A more complete set of loop transforming directives
  - TR11 includes reverse and interchange directives
  - Considering other transformations that include fission, fuse, collapse and nestify
  - Can transform generated loops using the apply clause
  - Considering characterizing loop-based work distribution constructs as transformations

- Clauses and directives to support generalized induction
  - Capture computation that follows a well-defined sequence across loop iterations
  - Generalizes behavior of linear clause and of loop iteration variables
  - Also strongly related to reductions, including addition of declare induction directive

- Efficient use of multiple compilation units (i.e., support for efficient IPO)?
- Support for pipelining, other computation/data associations; data-flow?
Many Changes for OpenMP 6.0 will extend 5.2 effort

- OpenMP ARB adopted OpenMP 5.2 on November 11, 2021 (131 passed issues)
- Large portions of specification now generated from JSON-based database
  - Section headers and directive and clause format
  - Cross references, index entries, hyperlinks and many other document details
  - Ensured syntax of directives and clauses is well-specified and consistent
  - Ensured restrictions are consistent and not just implied by syntax
  - Made C++ attribute syntax a first-class citizen
  - Deprecated one-off syntax choices, many other inconsistencies (12 new deprecation entries)
- Simplified combined and composite construct specification (hope to improve further)
- We will continue to migrate to specification of features in the database
  - Biggest remaining aspects are tools interfaces and runtime “property”
  - Will continue to improve readability and documentation of specification (e.g., “property“ entries)
  - Long-term plan will capture sufficient information in database to generate much more, including grammar, quick reference guide, and header and runtime library routine stub files
Some other improvements expected in OpenMP 6.0

- Removal of features that were deprecated in 5.0, 5.1 or 5.2
- Deeper support for descriptive and prescriptive control
  - Already adopted strict support for specifying `num_threads` prescriptively
  - Will likely extend to SIMD support
  - Will generalize mechanism to specify prescriptiveness of clauses and directives
- Wider use of C++ attribute syntax
  - Make C++ support “more C++-like”
    - Also clarified conditions for implicitly declared reduction operators for class types
  - Likely to include improvements for `threadprivate` and `declare target`
- Strengthening task-oriented execution changes begun in OpenMP 3.0
- Continuing to extend support for tool interfaces
- 67 issues already adopted in TR11, considering over 200 others for 6.0