The TBB functionality considered for removal in future versions

1 Contents

2 Motivation

We are planning to improve TBB through renewal and update with latest C++ standards to increase usability. To do this, we will need to evaluate removal of some TBB features in some future releases. Features under consideration are mapped to updated options as described in the documentation. Please take extra attention to section 4 in this document, and provide your feedback to us.

3 The functionality considered for removal

The following subsections list the functionality evaluating for removal, in addition to already mentioned functionality in TBB Developer Reference (Appendices: Compatbility Features) .

3.1 Pre C++11 compatibility API

Obsoleted functions are associated with C++03 and C++98. Any functionality aligned with these specifications will be updated to C++11.

The following table summarizes the TBB functionality that can be directly replaced with C++11.

TBB functionality	**Replacement**
tbb::atomic	std::atomic
tbb::flow::tuple (incl. helper classes)	std::tuple
tbb::mutex	std::mutex
tbb::recursive_mutex	std::recursive_mutex

tbb::critical_section (incl. tbb::improper_lock)	std::mutex
tbb::hash (incl. tbb::hasher)	std::hash
tbb::tbb_thread / std::thread / std::this_thread	std::thread with possible minimal changes related to std::chrono
std::lock_guard / std::unique_lock (incl. helper classes)	Minimal changes related to std::chrono might be required
std::condition_variable (incl. std::cv_status, std::timeout, std::no_timeout)	Minimal changes related to std::chrono might be required
tbb::tbb_exception / tbb::captured_exception / tbb::movable_exception	No more needed due to TBB exact exception propagation

3.2 PPL (Microsoft* Parallel Patterns Library) compatibility API

The following functionality was provided for compatibility with Microsoft* Parallel Patterns Library (PPL).

TBB functionality	Replacement
Concurrency::critical_section	std::mutex
Concurrency::reader_writer_lock (incl. Concurrency::improper_lock)	std::shared_mutex (*It will be provided by TBB in pre-C++17 environments)
Concurrency::parallel_invoke	tbb::parallel_invoke
Concurrency::parallel_for (first, last, f)	tbb::parallel_for (first, last, f)
Concurrency::parallel_for_each	tbb::parallel_for_each
Concurrency::task_group (incl. helper classes)	tbb::task_group
Concurrency::structured_task_group (incl. helper classes)	tbb::task_group/ tbb::structured_task_group

3.3 Redundant functionality

The following table summarizes the TBB functionality that significantly duplicates other existing functionality or have little practical usage

TBB functionality	Replacement

tbb::task_scheduler_init	tbb::task_arena, tbb::global_control (it will be extended to support the blocking terminate preview functionality)
tbb::pipeline (incl. tbb::filter, tbb::thread_bound_filter)	tbb::parallel_pipeline, tbb::flow::async_node
task priorities	Flow graph node priorities are already available. Dynamic arena-level priorities will be provided
tbb::flow::sender / tbb::flow::receiver / tbb::flow::continue_receiver	Remain as unspecified base types for flow graph classes
(preview) tbb::serial::parallel_for	Limit the number of threads to 1 with task_arena or global_control
(preview) runtime_loader (aka tbbproxy library)	No replacement is planned
Allocator template parameter for the flow graph nodes.	None.

4 The functionality considered for reworking

The following list contains the functionality considered for reworking. It might be removed, reworked or remained "as is" depending on evaluation feedback.

Functionality	Replacement
tbb::reader_writer_lock	std::shared_mutex (*It will be provided by TBB in pre-C++17 environments)
tbb::structured_task_group (incl. helper classes)	tbb::task_group
tbb::parallel_do	tbb::parallel_for_each (*it will be extended to support the feeder functionality)
tbb::aligned_space	std::aligned_storage
tbb::flow::source_node	tbb::flow::input_node, which is same as tbb::flow::source_node, but inactive by default. User needs to call method "activate()" explicitly.

5 Contacts

If you have any questions or concerns, please email Inteltbbdevelopers@intel.com.