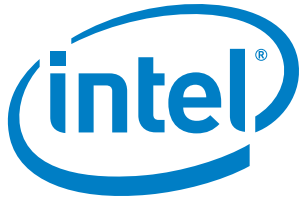


Intel® Threading Building Blocks (Intel® TBB) 2.2

In-Depth

Contents

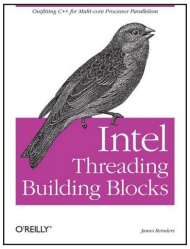
Intel® Threading Building Blocks (Intel® TBB) 2.2

Intel® Threading Building Blocks (Intel® TBB) is an award-winning C++ template library that abstracts threads to tasks to create reliable, portable, and scalable parallel applications. Use Intel TBB to implement task-based parallel applications and enhance developer productivity for scalable software on multicore platforms. Intel TBB is the most efficient way to implement parallel applications and unleash multicore platform performance compared with other threading methods like native threads and thread wrappers.

Productivity: Improves developer productivity by using task-based abstractions that make it easier to get scalable and reliable parallel applications with fewer lines of code (see figure 2). Task-based algorithms, containers, and synchronization primitives simplify parallel application development.

Future-proof applications: Application performance automatically improves as processor core count increases by using abstract tasks. Sophisticated task scheduler dynamically maps tasks to threads to balance the load among available cores, preserve locality, and maximize parallel performance.

Portability: Expand customer base by using a production-ready, open solution for parallelism that is available on a broad range of platforms. Available as a commercial and open source project, Intel TBB is coded in C++ and available on a multitude of platforms to provide a cross-platform solution for parallelism. Intel TBB is available as a standalone product, open source, or with the Intel® Compiler Professional Editions and Intel® Parallel Studio for a more complete and cost-effective solution.



Order the Intel® TBB Book at: http://www.amazon.com/Intel-Threading-Building-Blocks-Parallelism/dp/0596514808/ref=pd_bbs_sr_1?ie=UTF8&s=books&qid=1215536190&sr=1-1

Features

Intel TBB offers comprehensive, abstracted templates, containers, and classes for parallelism. Version 2.2 expands usage models and improves performance and usability. Figure 1 highlights the major functional groups within Intel TBB 2.2. Be sure to go to the What's New in 2.2 section on threadingbuildingblocks.org for a more detailed description of the new capabilities within Intel TBB 2.2.

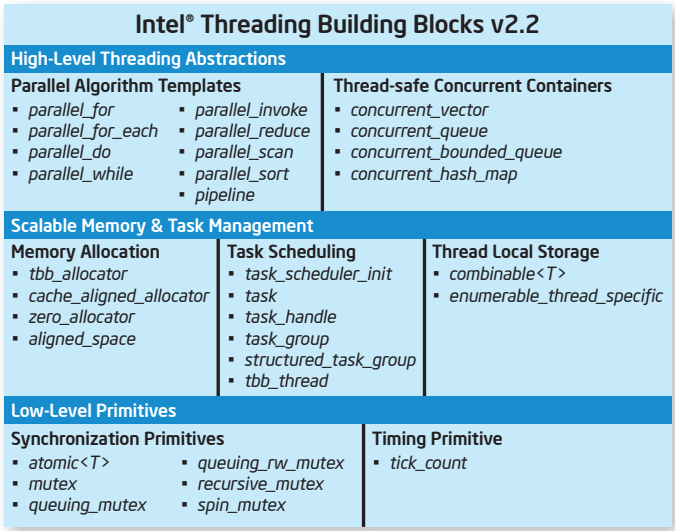


Figure 1: Intel TBB offers comprehensive, abstracted templates, containers, and classes for parallelism. Version 2.2 expands usage models and improves performance and usability.

Intel TBB lets developers focus on adding value to their application instead of thread management. Figure 2 highlights a dramatically simpler multithreaded implementation with Intel TBB versus native threads. Intel TBB utilizes robust functions that reduce threading errors like deadlock and race conditions.

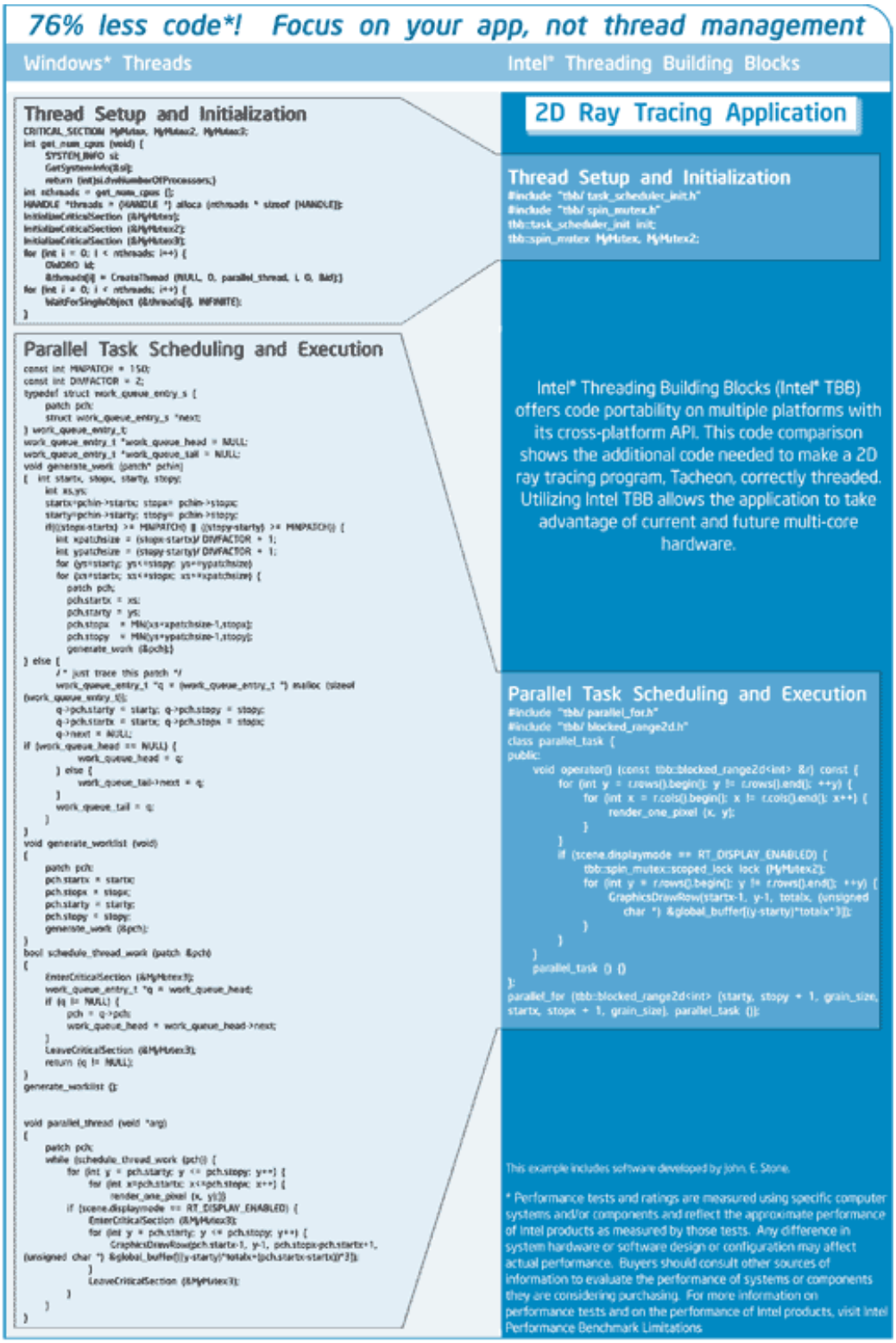


Figure 2: Side-by-side comparison of equivalent Windows* thread functionality that requires significantly more code to make a 2D ray tracing program, Tacheon, correctly threaded. Linux* and Mac OS* X developers can expect similar results.

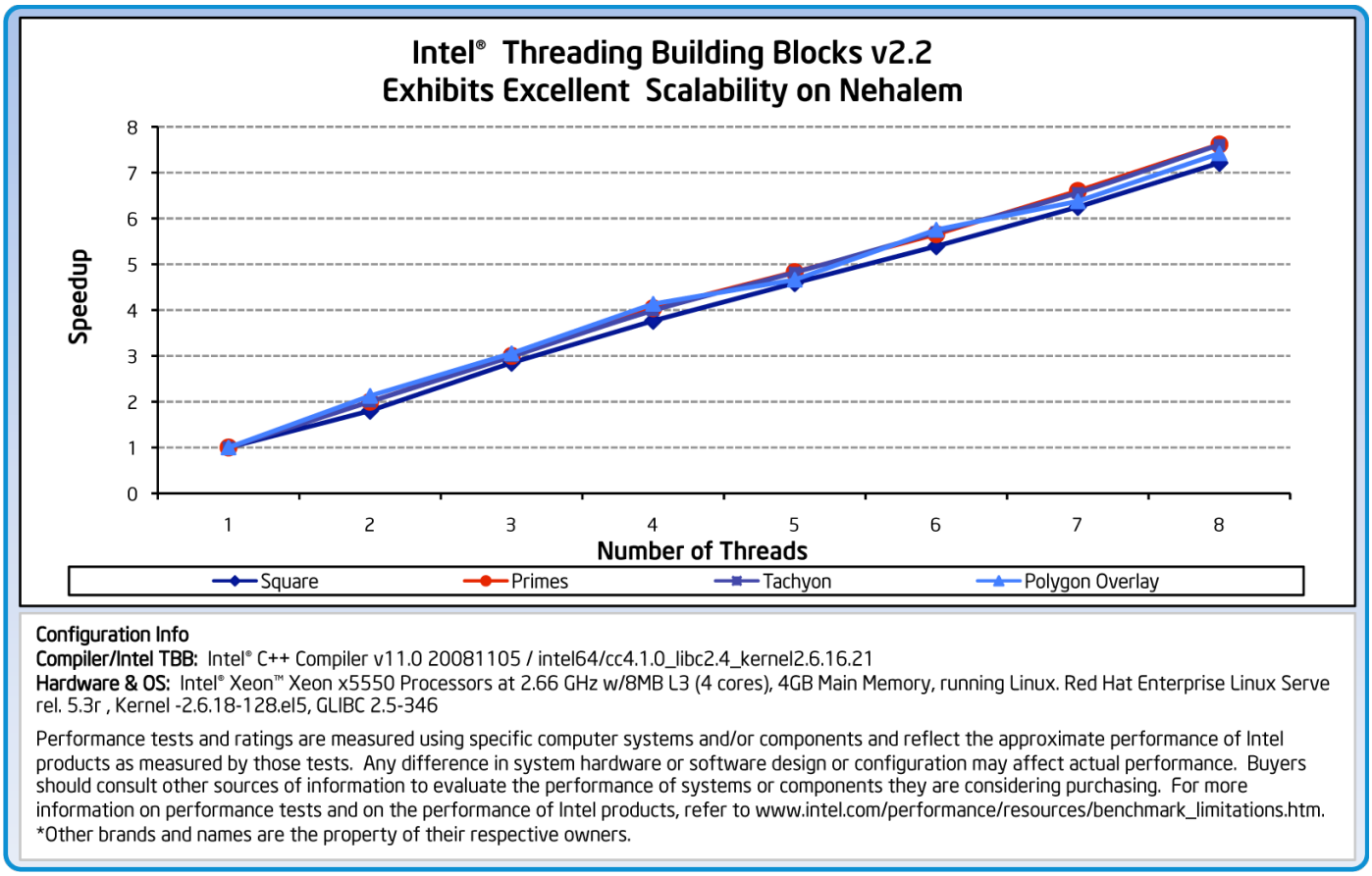


Figure 3: Excellent scalability and improved performance using Intel TBB versus a serial implementation. Linux and Mac OS X developers can expect similar results.

New in This Release

Intel TBB v2.2 offers several functionality, performance, and usability enhancements over 2.1.

Improved performance:

- Improved performance of scalable memory allocator
- Significant redesign of task scheduler for better performance and scalability
- Better performance of affinity partitioner
- auto_partitioner is the default for loop templates instead of simple_partitioner

New in scalable memory allocator:

- Intel® TBB 2.2 supports automatic replacement of OS allocator with its scalable memory allocator (Microsoft Windows* and Linux* OS)

Improvements in task scheduler:

- Intel® TBB 2.2 supports automatic initialization of task scheduler (task_scheduler_init is now optional)
- Support for task groups

New and improved parallel algorithms:

- New algorithms parallel_invoke and parallel_for_each
- New thread bound filter in pipeline
- Simplified interface for parallel_for for common loops
- Expanded support of Lambda expressions makes it easier to read and maintain code when using lambda capable compiler such as Intel® C++ Compiler 11.0 and above

New and improved data containers:

- New classes enumerable_thread_specific and combinable to support cross-platform thread local storage and algorithms for it
- Unbounded non-blocking interface for concurrent_queue and new blocking concurrent_bounded_queue
- Simplified interfaces for concurrent_hash_map
- Improved interfaces for concurrent_vector

Technical Support

Intel® Software Development Product purchases may include a year of support services, which provide access to Intel® Premier Support and all product updates during that time. Intel Premier Support gives you online access to technical notes, application notes, and documentation.

Which Intel TBB License is Right For Your Needs?

Intel TBB is available commercially as a binary distribution, and in open source in both source and binary forms. If you need commercial support services you should purchase either a standalone commercial license or take advantage of the considerable value in purchasing the Intel® Parallel Studio or Intel® Compiler Professional Edition. If your legal counsel is comfortable with your use of software under the Intel TBB open source license and you do not require commercial support services, please download the latest version of open source Intel TBB (http://threadingbuildingblocks.org). Finally, if you require the ability to modify or distribute the commercial source code of TBB, contact your Intel representative for more information.

When built from source, Intel TBB is intended to be highly portable, and so it supports a wide variety of operating systems and platforms. Binary distributions, including commercial distributions, are validated and officially supported for a variety of select hardware, software, operating systems, and compilers.

0809/BLA/CMD/PDF 321517-001