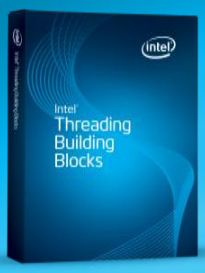


PARALLELIZE APPLICATIONS FOR PERFORMANCE

Intel® Threading Building Blocks 4.2

Product Brief

Top Features

- Rich set of components to efficiently implement higher-level, task-based parallelism
- Future-proof applications to tap current and future multicore and many-core platforms
- Compatible with multiple compilers and portable to various operating systems
- Available with open source or commercial license

Available in the following suites or standalone:

- Intel® Cluster Studio XE
- Intel® Parallel Studio XE
- Intel® C++ Studio XE
- Intel® Composer XE
- Intel® C++ Composer XE
- Standalone as Intel® Threading Building Blocks

OS Commercial Support:

- Windows*
- Linux*
- OS X*

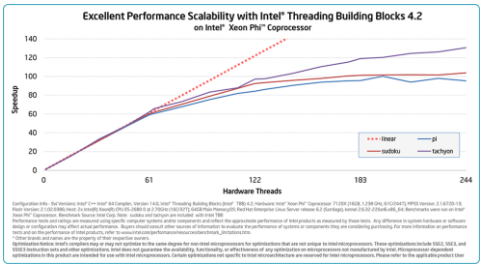
"Intel® TBB provided us with optimized code that we did not have to develop or maintain for critical system services. I could assign my developers to code what we bring to the software table—crowd simulation software."

Michaël Rouillé, CTO, Golaem

Simplify Parallelism with a Scalable Parallel Model

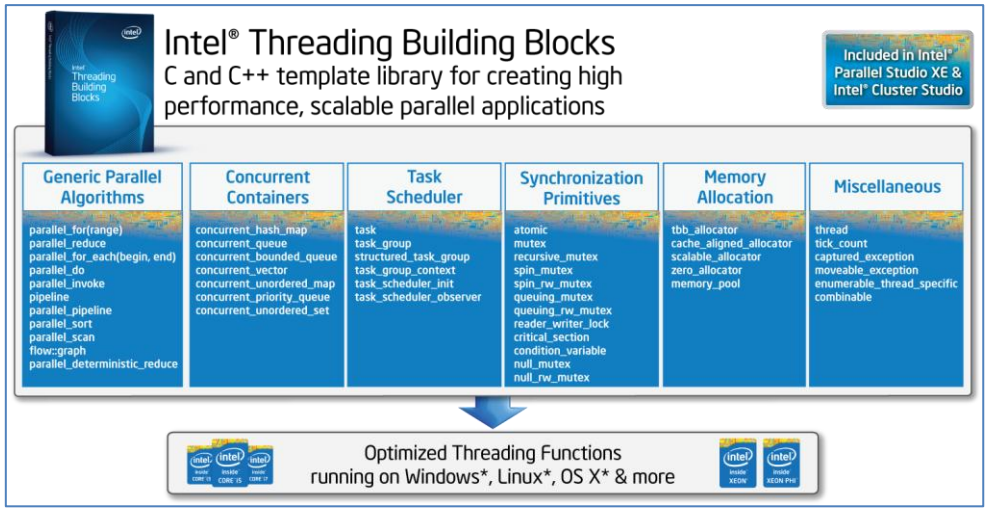
Intel® Threading Building Blocks (Intel® TBB) 4.2 is a widely used, award-winning C++ template library for creating high performance, scalable parallel applications. Intel TBB is the #1 choice of developers looking to implement cross-platform parallel applications. It automatically determines the best thread scheduling and work distribution to efficiently use the power and performance of multicore and many-core hardware.

- **Performance Scalability with Future-proofing** – Intel TBB provides a simple and rapid way of developing robust parallel applications that abstracts platform details and threading mechanisms for performance that scales with increasing core counts.



Intel® TBB yields linear scaling in these example applications running on Intel® Xeon® processors and Intel Xeon Phi™ coprocessors

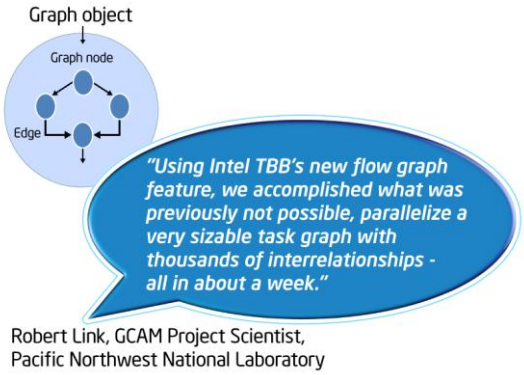
- **Productivity and Reliability** – Intel TBB provides abstractions that make it easier to write scalable and reliable parallel applications with fewer lines of code.



Intel® TBB Pre-Tested Capabilities

- **Compatible** – Compatible with multiple compilers and operating systems, Intel TBB fits within your environment making it easy to use and maintain.
- **Interoperable** – Multiple Intel TBB-based modules seamlessly interoperate in a user's application, helping avoid over subscription when other programming models are in use

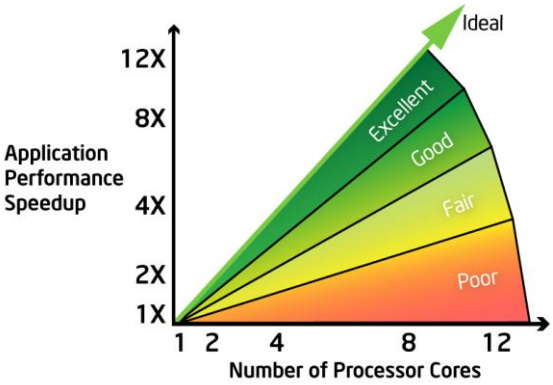
Top Features



Flow Graph

The flow graph feature provides a flexible and convenient API for expressing static and dynamic dependencies between computations. It is customizable for a wide variety of problems. It also extends the applicability of Intel® Threading Building Blocks (Intel® TBB) to event-driven/reactive programming models.

Intel TBB delivers high performing and reliable code with less effort than hand-made threading. Pre-tested algorithms, concurrent containers, synchronization primitives, and a scalable memory allocator simplify parallel application development.



Design For Scaling

Dynamic Task Scheduler

Application performance can automatically improve as processor core count increases by using abstract tasks. The sophisticated Intel® TBB task scheduler dynamically maps tasks to threads to balance the load among available cores, preserve cache locality, and maximize parallel performance. The implementation supports C++ exceptions, task/task group priorities, and cancellation which are essential for large and interactive parallel C++ applications.

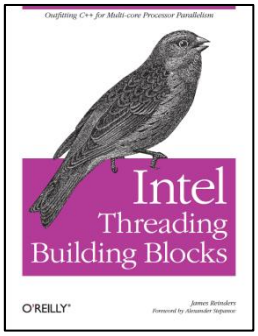
Dynamic task scheduler and parallel algorithms support nested and recursive parallelism as well as running parallel constructs side-by-side. This is useful for introducing parallelism gradually and helps independent implementation of parallelism in different components of an application.

```
tbb::parallel_for (0, n, [](int i) {
    #pragma simd reduction(+:S[i])
    for( int j=0; j<n; ++j )
        S[i] += A[i][j];
});
//No OS specific code required
```

Cross Platform Support and Composability

Organizations that require cross platform support today or anticipate needing it in the future should consider Intel TBB. It is validated and commercially supported on Windows*, Linux*, and OS X* platforms, using multiple compilers. It is also available on FreeBSD*, IA-based Solaris*, and PowerPC*-based systems via the open source community. Intel TBB is optimized for multicore architectures and Intel® Xeon Phi™ coprocessor.

Intel TBB is designed to co-exist with other threading packages and technologies. Different components of Intel TBB can be used independently and mixed with other threading technologies.



Order the Intel® Threading Building Blocks book online at amazon.com

Top Community Support

Broad support from an involved community provides developers access to additional platforms and OS's. Intel® Premier Support services and Intel® Support Forums provide confidential support, technical notes, application notes, and the latest documentation.

A complete documentation package and code samples are readily available both as a part of Intel TBB installation and online at http://threadingbuildingblocks.org. The User Guide provides an introduction into Intel TBB. The Design Patterns chapter in the User Guide covers common parallel programming patterns and how to implement them using Intel TBB. The Reference Manual contains formal descriptions of all classes and functions implemented in Intel® TBB.

Additional components for Performance and Productivity

Parallel Algorithms Generic implementation of common patterns	Generic implementations of parallel patterns such as parallel loops, flow graphs, and pipelines can be an easy way to achieve a scalable parallel implementation without developing a custom solution from scratch.
Concurrent Containers Generic implementation of common idioms for concurrent access	Intel® Threading Building Blocks (Intel® TBB) concurrent containers are a concurrency-friendly alternative to serial data containers. Serial data structures (such as C++ STL containers) often require a global lock to protect them from concurrent access and modification; Intel TBB concurrent containers allow multiple threads to concurrently access and update items in the container increasing allowed concurrency and improving an application's scalability.
Synchronization Primitives Exception-safe locks, condition variables, and atomic operations	Intel TBB provides a comprehensive set of synchronization primitives with different qualities that are applicable to common synchronization strategies. Exception-safe implementation of locks helps to avoid a dead-lock in programs which use C++ exceptions. Usage of Intel TBB atomic variables instead of the C-style atomic API minimizes potential data races.
Scalable Memory Allocators Scalable memory manager and false-sharing free memory allocator	The scalable memory allocator avoids scalability bottlenecks by minimizing access to a shared memory heap via per-thread memory pool management. Special management of large (≥8KB) blocks allows more efficient resource usage, while still offering scalability and competitive performance. The cache-aligned memory allocator avoids false-sharing by not allowing allocated memory blocks to split a cache line.
Create arbitrary task trees	When an algorithm cannot be expressed with high-level Intel TBB constructs, the user can choose to create arbitrary task trees. Tasks can be spawned for better locality and performance or en-queued to maintain FIFO-like order and ensure starvation-resistant execution.
Conditional Numerical Reproducibility	Ensure deterministic associativity for floating-point arithmetic results with the new Intel TBB template function 'parallel_deterministic_reduce'.
C++11 Support	Intel TBB can be used with C++11 compilers and supports lambda expressions. For developers using parallel algorithms, lambda expressions reduce the time and code needed by removing the requirement for separate objects or classes.

Select the right Intel® Threading Building Blocks (Intel® TBB) license

- **Commercial Binary Distribution** for customers who may require commercial support services. Attractive pricing available for academic, student and classroom usage.
- **Open Source Distribution** can be used under GPLv2 with the runtime exception allowing usage in proprietary applications. Allows support for additional OSs and hardware platforms. Both source and binary forms are available for download from http://threadingbuildingblocks.org.
- **Custom license** available if you require the ability to modify or distribute the commercial source code of Intel TBB. Contact your Intel representative for more information.

What's New in version 4.2

Feature	Benefit
Support for Latest Intel Architectures	Take advantage of the newest features in Intel's latest processors including Transactional Synchronization Extensions (TSX). Adds support for Intel® Xeon Phi™ coprocessor for Windows and Intel® Xeon™ Processor (Ivy Bridge-EP). Selecting the best models for your application today will set a path for you to take full advantage of multicore and many-core performance without re-writing your code. Start today by implementing parallelism for today's architecture and be ready for future architectures.
Lower memory overhead	Improved heuristics in the memory allocator reduce memory overhead by intelligently releasing unused or stale memory.
Improved handling of large memory requests	Improved handling of large (>8K-128MB) memory requests results in better performance when using frequent large memory allocations. Use of big memory pages can now be explicitly enabled via a function call or environment variable.
Better Fork Support	Fork safety through a user enabled API that ensures Intel TBB worker threads are completed before executing a fork.
PPL* Compatibility	Improved compatibility with Parallel Patterns Library (PPL) by adding concurrent_unordered_multimap and concurrent_unordered_multiset API's.
Windows* Store	Customers that use Intel TBB in their applications can now submit and sell their app through the Windows Store.

Purchase Options: Language Specific Suites

Several suites are available combining the tools to build, verify and tune your application. The product covered in this product brief is highlighted in blue. Named-user or multi-user licenses along with volume, academic, and student discounts are available.

	Suites >>	Intel® Cluster Studio XE	Intel® Parallel Studio XE	Intel® C++ Studio XE	Intel® Fortran Studio XE	Intel® Composer XE	Intel® C++ Composer XE	Intel® Fortran Composer XE
Components	Intel® C / C++ Compiler	●	●	●		●	●	
	Intel® Fortran Compiler	●	●		●	●		●
	Intel® Integrated Performance Primitives[3]	●	●	●		●	●	
	Intel® Math Kernel Library[3]	●	●	●	●	●	●	●
	Intel® Cilk™ Plus	●	●	●		●	●	
	Intel® Threading Building Blocks	●	●	●		●	●	
	Intel® Inspector XE	●	●	●	●			
	Intel® VTune™ Amplifier XE	●	●	●	●			
	Intel® Advisor XE	●	●	●	●			
	Static Analysis	●	●	●	●			
	Intel® MPI Library	●						
	Intel® Trace Analyzer & Collector	●						
	Rogue Wave IMSL* Library[2]							●
	Operating System[1]	W, L	W, L	W, L	W, L	W, L	W, L, O	W, L, O

Note: [1] Operating System: W=Windows*, L= Linux*, O= OS X*. [2] Available in Intel® Visual Fortran Composer XE for Windows with IMSL*
[3] Not available individually on OS X, it is included in Intel® C++ & Fortran Composer XE suites for OS X

Technical Specifications

Specs at a Glance	
Processor Support	Validated for use with multiple generations of Intel and compatible processors including but not limited to: Intel® Xeon™ Processor, Intel® Core™ processor family, Intel® Atom™ processor family and Intel® Xeon Phi™ coprocessor.
Operating Systems	Use the same API for application development on multiple operating systems: Windows*, Linux* and OS X*.
Development Tools and Environments	Compatible with compilers from vendors that follow platform standards (e.g., Microsoft*, GCC, Intel). Can be integrated with GNU* tools Microsoft Visual Studio* 2008, 2010 and 2012.
Programming Languages	Natively supports C++ development; cross language usage examples provided for C#/.NET.
System Requirements	Refer to www.intel.com/software/products/systemrequirements/ for details on hardware and software requirements.
Support	All product updates, Intel® Premier Support services and Intel® Support Forums are included for one year. Intel Premier Support gives you secure, web-based, engineer-to-engineer support.
Community	Share experiences with other users of Intel® TBB and other parallel programming tools at the Intel moderated forum: http://software.intel.com/en-us/forums/.

Learn more about Intel TBB

- Click or enter the link below: http://intel.ly/intel-tbb
- Or scan the QR code on the left

Download a free 30-day evaluation

- Click or enter the link below: http://intel.ly/sw-tools-eval
- Click on 'Performance Libraries' link

Optimization Notice Notice revision #20110804

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.