

コンパイラーとスレッド化ライブラリー

インテル® Parallel Composer 2011

製品紹介

インテル® Parallel Composer 2011



「Trading Systems Lab では、インテル® Parallel Studio の C++ コンパイラーを使用することにより、TSL Algo Auto-Design Platform で使用されるマルチモードの取引シミュレーターのパフォーマンスを 10% から 20% 向上することができました。Microsoft® Visual C++ との互換性は素晴らしく、Parallel Studio でより多くの並列化機能を使用するのを楽しみにしています。」

Trading Systems Lab 社
代表取締役
Mike Barna 氏

マルチコアシステム向け C/C++ シリアルおよび並列アプリケーションのビルド

インテル® Parallel Composer 2011 は、Windows* ベースのクライアント・アプリケーションで最高のパフォーマンスを実現するためのインテル® C++ コンパイラー、パフォーマンス・ライブラリー、並列開発モデル、デバッグ機能を備えた強力な開発ツールセットです。Microsoft® Visual Studio* 2005、2008、および 2010 に統合され、Microsoft® Visual C++ との互換性を保ち、使い慣れた環境で作業できるだけでなく、これまでの投資を無駄にしません。

インテル® Parallel Composer は単体製品として、あるいはインテル® Parallel Studio として購入することができます。インテル® Parallel Studio には、次の製品も同梱されています。

- 革新的なスレッド化アシスタント
インテル® Parallel Advisor は、既存アプリケーションの並列化実装を支援します。
- メモリーエラーとスレッド化エラーチェッカー
インテル® Parallel Inspector は、メモリーリークなどのメモリーエラーを分析し、スレッドの正当性を検証してスレッドのデバッグを支援します。
- スレッドとパフォーマンス・プロファイラー
インテル® Parallel Amplifier は、マルチコア・パフォーマンスのボトルネックを簡単に素早く発見します。

インテル® Parallel Composer のコンポーネント

- 32 ビット・プロセッサ用インテル® C++ コンパイラーと、32 ビット・システムにおいて 64 ビット・アプリケーションを作成するクロスコンパイラー
- インテル® Parallel Debugger Extension。Microsoft® Visual Studio* デバッガーに統合されます。
- インテル® Parallel Building Blocks。さまざまな並列化実装方法を提供します。含まれるセットは次のとおりです。

- インテル® スレディング・ビルディング・ブロック (インテル® TBB)。タスクベースの並列化の実装に使用します。インテル® TBB は、マルチコア・プラットフォームでスケラブルかつハイパフォーマンスなクロスプラットフォーム・ソフトウェアを開発する際の生産性を向上します。オープンソース版も用意されています。

- インテル® Cilk™ Plus。C/C++ 言語拡張です。ループおよびタスクの並列化と、コンパイラーによるプログラミング・モデルを適用することで、C/C++ コードに並列化を素早く実装できます。

- インテル® Array Building Blocks (ベータ版)。データ並列数値計算アプリケーションでデータの並列化を実装し、マルチコア・プラットフォームでスケーラブルかつハイパフォーマンスなクロスプラットフォーム・ソフトウェアを開発する際の生産性を向上します。

■ インテル® インテグレートッド・パフォーマンス・プリミティブ (インテル® IPP)。インテル® IPP は、マルチメディア、データ処理、基本的な算術関数、通信アプリケーション向けに高度に最適化された、関数群からなる広範囲なマルチコア対応ライブラリーです。インテル® IPP には、手動で最適化されるプリミティブ・レベルの関数と、コーデックなどのハイレベルなスレッド・ソリューションの両方が含まれています。インテル® C++ コンパイラーだけでなく、Visual C++ 環境および .NET 環境でも使用できます。インテル® マス・カーネル・ライブラリー (インテル® MKL) を別途購入すると、より高度な算術関数も利用できます。

■ サンプルコードと入門ガイド。ツールをすぐに使用できるように用意されています。

入門ガイドと拡大する並列化のコミュニティとの交流

インテル® Parallel Composer には、入門ガイドで紹介されているサンプルコードも含まれています。また、Web サイトには入門ガイドで紹介されている短編ビデオもあります。

入門ガイドには、Web サイトや Visual Studio* の [ヘルプ] メニュー (詳細なドキュメントも提供)、Windows* の [スタート] ボタンから表示されるインテル® Parallel Studio またはインテル® Parallel Composer のメニューからアクセスできます。さらに、インストールの完了時に表示されるプロンプトからも利用可能です。並列化は初めての方も、熟練者の方も、入門ガイドにひととおり目を通すことをお勧めします。

ツールを使用し始めたら、開発者コミュニティが役立つでしょう。インテルのマルチコア・プロセッサ・ベースのシステムを活用するために、拡大するコミュニティに是非ご参加ください。インテルでは、開発者が活発にアイデアを出し合い、意見を投稿し、ポイントを獲得して、インテル® Black Belt Software Developer の称号を手にするフォーラムを提供しています。また、開発者の関心を引くような並列化に関するさまざまなトピックを擁するブログ、ダウンロード、大規模なナレッジベースも用意されています。並列プログラミングおよびマルチコア・コミュニティについては、<http://software.intel.com/en-us/articles/intel-parallel-studio/> (英語) を参照してください。

インテル® C++ コンパイラー :

Microsoft* Visual Studio* 統合と互換性

インテル® Parallel Composer の全機能は、Microsoft* Visual Studio* 2005、2008、および 2010 にシームレスに統合されます。これにより、Microsoft* Visual Studio* のツールや設定が変更されることはありません。インテル® Parallel Composer の機能は、オプションを設定するのと同じように簡単に使用できます。これらの機能は、インテル® Parallel Composer をインストールすると Visual Studio* のツールバーや [プロジェクト] メニューなどの特定のプルダウンメニューに追加されます。ソリューションまたはプロジェクト・ファイルを右クリックするか、プロパティのポップアップ・ウィンドウから、ソリューションやプロジェクトで使用するコンパイラー (インテル® C++ コンパイラーまたは Visual C++ コンパイラー) を指定したり、簡単に切り替えることができます。このコンパイラーは、ネイティブ 32 ビット開発環境とクロスコンパイル環境 (32 ビット・ホストで 64 ビット・アプリケーションを作成) を提供します。32 ビット環境のみ、または 64 ビット環境のみ、あるいは両環境のインストールを選択できます。

インテル® C++ コンパイラーは Visual C++ とバイナリー互換で、多くの場合、インテル® C++ コンパイラーでビルドすることでモジュールやアプリケーション全体のパフォーマンスが大幅に向上します。また、インテル® Parallel Debugger Extension により、並列コードのデバッグを迅速に行うこともできます。インテル® C++ コンパイラーは、開発者に多くの利点があります。インテル® Parallel Composer のほかのコンポーネントや、インテル® Parallel Studio をすべて使用する必要はありませんが、これらとともに使用することでより多くの利点が得られます。例えば、インテル® TBB とインテル® IPP を Visual C++ コンパイラーとインテル® C++ コンパイラーで使用できます。また、インテル® Parallel Inspector のメモリーリークや並列性チェック機能を、Visual C++ コンパイラーとインテル® C++ コンパイラーでビルドされたアプリケーションで使用することも可能です。つまり、必要に応じて Visual C++ コンパイラーも使用しながら、互換性を持つインテル® C++ コンパイラーを含む、インテル® Parallel Studio スイートのさまざまな活用方法を検討する理由がここにあります。

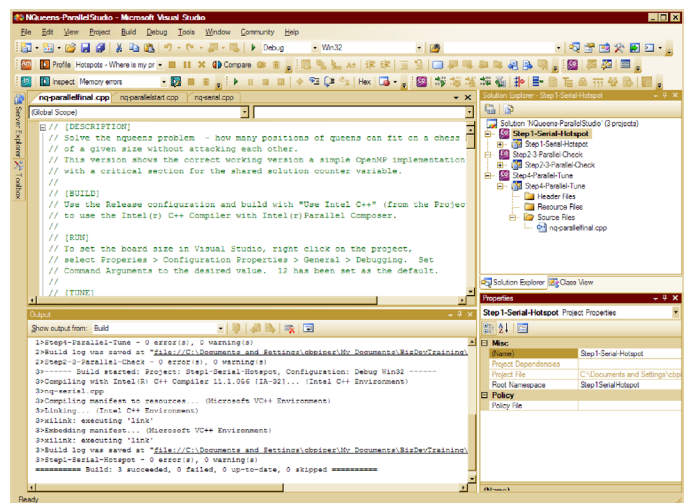


図 1: インテル® Parallel Composer は Microsoft* Visual Studio* に統合されます。ここでは、『インテル® Parallel Studio 入門ガイド』で使用されているサンプルコードのソリューションを表示しています。

インテル® Parallel Building Blocks

インテル® TBB

インテル® スレディング・ビルディング・ブロック (インテル® TBB) は、一般的な並列化を有効にするための C++ テンプレート・ライブラリーです。スケーラブルなメモリー割り当て、負荷分散、ワークスチール・タスク・スケジューリング、スレッドセーフなパイプラインとコンカレント・コンテナ、高度な並列アルゴリズム、さまざまな同期プリミティブが含まれます。インテル® TBB は、汎用のループとタスクを使用して並列アプリケーションを作成する C++ 開発者向けのライブラリーです。

インテル® TBB によりスレッド化プログラミングの 3 つの主要な課題を解決：

- **生産性**：並列化の実装を簡素化
- **正当性**：並列の同期問題の排除を支援
- **保守**：現在だけではなく、将来にも対応できるアプリケーションの作成を支援

インテル® TBB の利点：

- **将来も安心のアプリケーション**：インテル® TBB の洗練されたタスク・スケジューラーを使用することで、コア数（およびスレッド数）が増加するにつれ、アプリケーションが高速化します。
- **移植性**：並列化をいったん実装してしまえば、複数のプラットフォームでスレッドコードを実行できます。
- **相互運用**：多種多様なスレッド化手法、ハードウェア、オペレーティング・システムで動作します。

活発なオープンソース・コミュニティ：インテル® TBB には、オープンソース版もあります。 opentbb.org (英語) は、フォーラム、ブログ、コードサンプルなどが用意されている役立つ Web サイトです。

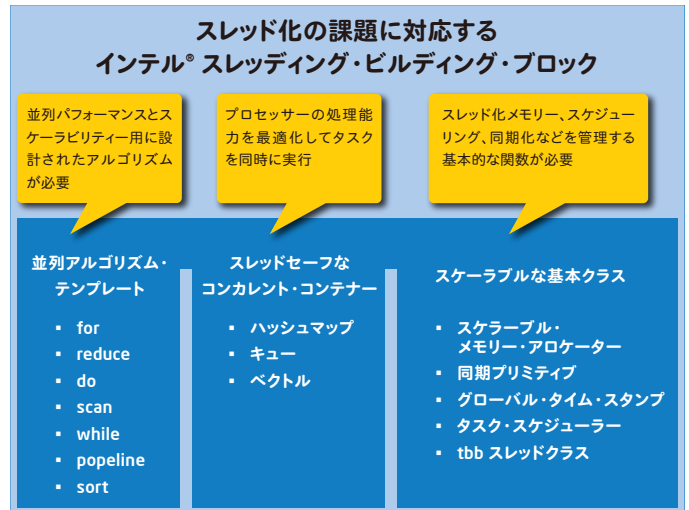


図 2: インテル® TBB の主な関数グループ

インテル® TBB は、スレッド化作業に伴うさまざまな問題を解決します。図 3 は、インテル® TBB で対応可能なスレッド化における 3 つの一般的な問題を示しています。

問題	解決方法
スレッド化を簡単に追加する方法	インテル® TBB の <code>parallel_for</code> コマンド <ul style="list-style-type: none">▪ 簡単な <code>for/next</code> ループの置換で、スレッド化の利点を活用▪ 決まった回数の独立したループ反復による負荷分散された並列実行
最良のスケーラビリティを得るためのスレッドの管理	インテル® TBB タスク・スケジューラー <ul style="list-style-type: none">▪ スレッドプールを管理し、ネイティブスレッドの複雑性を回避▪ 並列プログラミングにおける一般的なパフォーマンスの問題に対応するよう設計<ul style="list-style-type: none">- オーバーサブスクリプション：ハードウェア・スレッドごとに 1 つのスケジューラー・スレッド- 高いオーバーヘッド：プログラマーはスレッドではなくタスクを指定- ロード・インバランス：ワークスチールにより負荷を調整
並列環境におけるボトルネックのメモリー割り当て	インテル® TBB は、スレッドごとのメモリー管理アルゴリズムに基づいて、テスト、チューニングされたスケーラブルなメモリー・アロケーターを提供 <ul style="list-style-type: none">▪ STL テンプレート・クラスへのアロケーター引数として▪ <code>malloc/realloc/free</code> 呼び出しの代替として (C プログラム)▪ グローバルな <code>new</code> 演算子と <code>delete</code> 演算子の代替として (C++ プログラム)

図 3: インテル® TBB により並列化の 3 つの主要な問題に対応

インテル® Cilk™ Plus

インテル® Cilk™ Plus は、インテル® C/C++ コンパイラ固有の並列化の実装です。ベクトル化機能と、高度なループベースのデータ並列処理およびタスク処理を組み合わせることで、優れた機能を提供します。インテル® Cilk™ Plus は、単純なループ、データ、タスクを使用して並列アプリケーションを作成する C++ 開発者向けです。

インテル® Array Building Blocks

インテル® Array Building Blocks(ベータ版) は、高度なランタイム・ライブラリーを備えた API です。特定の並列メカニズムやハードウェア・アーキテクチャーに依存しない、データ並列プログラミングの汎用ソリューションを提供します。保守が簡単なハイレベルのアプリケーション指向の計算処理から、スケラブルで移植性が高く決定性のある並列実装を生成します。インテル® Array Building Blocks は、データ並列数値計算アルゴリズムを記述する C++ 開発者向けです。ベータ版 インテル® Array Building Blocks は、<http://software.intel.com/en-us/data-parallel/> (英語) から利用できます。

インテル® インテグレートッド・パフォーマンス・プリミティブ (インテル® IPP)

インテル® Parallel Composer には、インテル® IPP が含まれています。インテル® IPP は、マルチメディア、データ処理、通信アプリケーション向けに高度に最適化された、関数群からなる広範囲なマルチコア対応ライブラリーです。画像処理、画像圧縮、データ圧縮、ビデオコーディング、ストリング処理、暗号化、信号処理、オーディオコーディング、音声コーディング、音声認識、コンピューター・ビジョン、イメージカラー変換、ベクトル/行列の演算などの分野で頻繁に使用される基本的なアルゴリズムを含む、最適化された関数を多数提供します。別売のインテル® マス・カーネル・ライブラリー (インテル® MKL) を使用すると、より高度な算術関数も利用できます。

インテル® IPP には、手動で最適化されるプリミティブ・レベルの関数とコーデックなどのハイレベルなスレッド化されたサンプルの両方が含まれ、Visual C++ 環境と .NET 環境の両方で使用することが可能です。これらの関数やサンプルはすべてスレッドセーフで、多くは内部的にスレッド化され、今日のマルチコア・プロセッサを活用し、将来のメニーコア・プロセッサにスケラリングします。

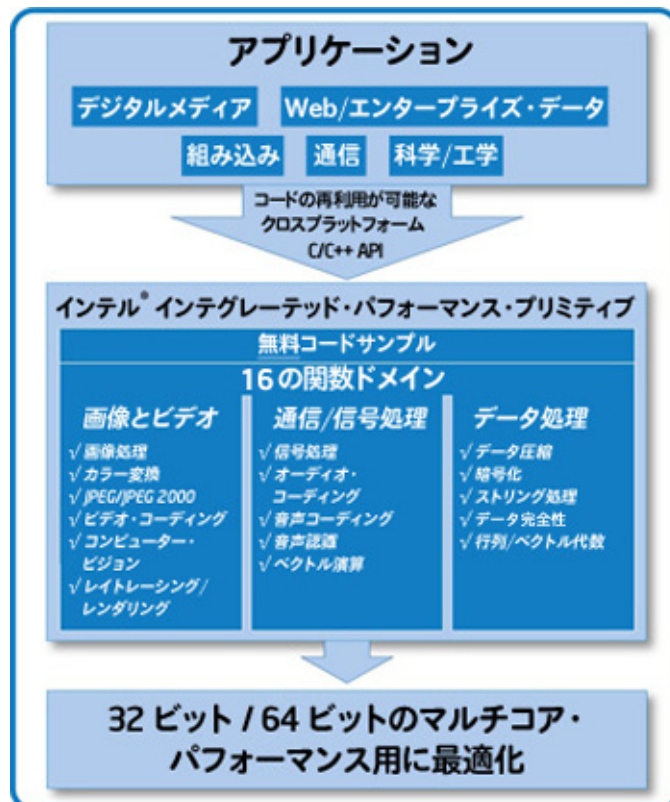


図 4: インテル® IPP はインテル® Parallel Composer に含まれ、上記の多様なドメインにおいてスレッド化済みのスレッドセーフなライブラリー関数を提供します。

インテル® IPP のパフォーマンス

アプリケーションとワークロードに応じて、インテル® IPP 関数は、同等のコンパイル済み C コードと比べ、何倍も高速に動作します。図 5 に示す画像のサイズ変更の例では、C++ ベースのコードを実行した場合は 338 マイクロ秒かかり、インテル® IPP の画像処理関数を使用して実行すると、111 マイクロ秒しかかからなかったことを示しています。つまり、パフォーマンスが 300% ほど向上しています。

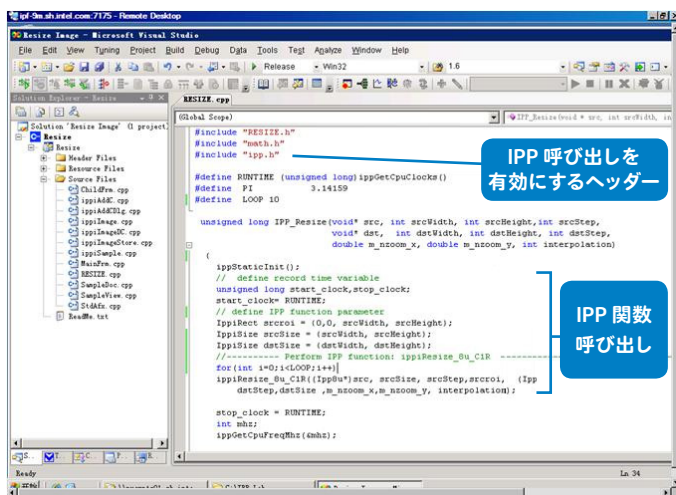


図 5: インテル® IPP ライブラリー呼び出しを Microsoft* Visual Studio* コードに簡単に統合

C++ プロジェクトのほか、インクルードされたラッパークラスを使用し、文字列処理、イメージ処理、信号処理、カラー変換、暗号化、データ圧縮、JPEG、行列、ベクトル演算のドメインにおいて、C# からインテル® IPP 関数への呼び出しをサポートすることで、C# プロジェクトでも使用できます。

この画像サイズの変更例では、インテル® IPP コードはコンパイル済み C++ コードよりも 3 倍高速に実行



図 6: この画像サイズの変更例 (256x256 ビットから 460x332 ビット) では、インテル® IPP を使用したアプリケーションが 111 マイクロ秒であるのに対し、C++ ベースのコードでは 338 マイクロ秒を要しています (システム構成: インテル® Xeon® プロセッサ、2.9 GHz、2 プロセッサ、4 コア/プロセッサ、2 スレッド/プロセッサ)

驚異的並列ループの最適化

独立した反復を持つデータ並列アルゴリズムでは、ループで「驚異的並列」コードを実現します。インテル® Parallel Composer は、そのようなループのパフォーマンスを最小限の労力で、最大限に引き出す 3 つの手法 (自動ベクトル化、インテルにより最適化された valarray コンテナの使用、自動並列化) をサポートしています。インテル® Parallel Composer は自動ベクトル化に適したループを自動的に検出します。これには、静的配列または動的配列を持つ明示的な for ループ、ベクトルコンテナと valarray コンテナ、または明示的なループを持つユーザー定義の C++ クラスなどがあります。例外として、暗黙的な valarray ループは、自動ベクトル化を行うか、あるいは最適化されたインテル® IPP ライブラリー・プリミティブを起動するよう指定することが可能です。自動ベクトル化と最適化された valarray ヘッダーの使用で、アプリケーションのパフォーマンスを最適化し、ストリーミング SIMD 拡張命令 (SSE) 対応のプロセッサを十分に活用することができます。

自動並列化では、並列で安全に実行できるループを検出し、自動的にマルチスレッド・コードを生成して、マルチコア・プロセッサを活用することで、アプリケーション・パフォーマンスが向上します。反復のパーティショニング、データの共有、スレッドのスケジューリング、および同期に関する低いレベルを処理して、ユーザーの負担を軽減します。

自動並列化は、自動ベクトル化と最適化された valarray ヘッダーの使用を補い、SSE 対応のマルチコアシステム上で最適なパフォーマンスを得られるようにします。マルチスレッド・アプリケーションのサポートに関する詳細は、ユーザーガイド (<http://software.intel.com/en-us/intel-parallel-composer/> (英語) でドキュメントのリンクをクリック) を参照してください。

本リリースで新しく追加されたガイド付き自動並列処理 (GAP) は、分析機能を提供する自動ベクトル化の拡張です。GAP 分析の結果はビルドログに表示され、ソースコードを変更することで、コンパイラが自動ベクトル化や自動並列化を安全に適用して、アプリケーションを最適化できる場合には、アドバイスが提供されます。

インテル® Parallel Debugger Extension

インテル® Parallel Composer には、インテル® Parallel Debugger Extension が含まれています。デバッガーには、Microsoft® Visual Studio® の [デバッグ] プルダウン・メニューからアクセスできます。インテル® C++ コンパイラーの拡張機能をサポートし、Visual C++ コンパイラーでコンパイルしたコードのデバッグにも使用できます。これは、Microsoft® Visual Studio® デバッガーの拡張であって、代替ではありません。

インテル® Parallel Debugger Extension は、並列アプリケーションの共有データやデータの依存関係について、詳しい洞察やアクセスを提供します。開発サイクルを短縮し、重大なランタイムエラーを引き起こす可能性のある潜在的なデータアクセス競合を早めに検出します。

共有データイベント検出や関数の再帰性検出などのインテル® Parallel Debugger Extension の高度な機能を活用するには、インテル® コンパイラーでデバッグ情報をインストールする /debug:parallel オプションを使用してコードをコンパイルしてください。

詳細は、インテル® Parallel Debugger Extension のホワイトペーパー (<http://software.intel.com/en-us/articles/parallel-debugger-extension/> (英語)) を参照してください。

システム要件

Microsoft® Visual Studio® 2005、2008、および 2010 (Express Edition は除く)

最新のシステム要件については、次の Web ページを参照してください。

www.intel.com/software/products/systemrequirements/ (英語)

サポート

インテル® Parallel Studio 製品をご購入いただくと 1 年間のサポートサービスが提供されます。このサポートには、インテル® プレミアサポートへの問い合わせ、製品の最新版のダウンロードおよび、テクニカル・ドキュメントへのアクセスが含まれています。

詳細は、次のサイトを参照してください。

<http://software.intel.com/en-us/articles/intel-parallel-studio/> (英語)

評価版のダウンロード

評価版のダウンロードについての詳細は、次の Web サイトを参照してください。

<http://www.intel.co.jp/jp/software/products/intel-parallel-studio-home/>

機能概要

- インテル® Parallel Composer は、Visual Studio* に統合され、投資を確保しながら、アプリケーションのパフォーマンスを最大限に引き出すための多数の機能を提供します。Visual Studio* の機能を変更することなく、既存の機能を拡張します。
- インテル® C++ コンパイラーは、インテル® Parallel Composer の一部で、Microsoft* Visual C++ と互換性があります。
- インテル® Parallel Debugger Extension は、Microsoft* デバッガーと統合され、スレッド化の問題を発見し、対処できるように Visual Studio* を強化します。これにより、アプリケーションの開発期間が短縮されます。
- インテル® Parallel Composer には、簡単な並列関数、並列データ配列、そして多数のスレッド化ライブラリー関数が含まれており、スレッド化作業を簡素化し、アプリケーション開発を促進します。
- 新機能のガイド付き自動並列処理を含む、自動並列化と自動ベクトル化は、インテル® Parallel Studio の一部で、開発を容易にして時間を短縮します。
- インテル® Parallel Building Blocks は、タスクやデータを並列化するための利便性の高い並列関数を備え、並列アプリケーションのより迅速な開発をサポートし、パフォーマンスの高いシリアル・アプリケーションの開発を容易にします。この製品には、インテル® TBB、インテル® Cilk™ Plus、およびインテル® ArBB (ベータ版) が含まれています。さらに、統合されたアレイ・ノテーションによる配列表記、データ並列に対応するインテル® IPP 関数も含まれているため、オーディオ、ビデオ、信号分析やその他のアプリケーション・クラスを高速処理できます。
- コードサンプルや入門ガイドなど、さまざまなドキュメントが用意されています。
- 開発者コミュニティでは、多くの開発者がコードの並列化に取り組んでいます。他の開発者の経験から学び、そしてご自身の知識や経験を共有してください。さまざまな特典も用意されています。
- インテル® Parallel Composer は、インテル® Parallel Studio の一部としても利用できます。インテル® Parallel Studio には、既存アプリケーションの並列化のアドバイスを提供するインテル® Parallel Advisor、メモリーリークやスレッドの正当性などのデバッグを支援するインテル® Parallel Inspector、そしてアプリケーションのパフォーマンスをさまざまな角度から測定するのに役立つプロファイラーを提供するインテル® Parallel Amplifier も含まれています。

究極のオールインワン・パフォーマンス・ツールキット：インテル® Parallel Studio 2011

今日のシリアル・アプリケーション、そして明日のソフトウェア・イノベーションのための設計

インテルは、シリアル・アプリケーションおよびマルチコアやメニーコア向けの新しいスレッド・アプリケーション用に設計された生産性ソリューション・セットにより、Microsoft* Visual Studio C++ 開発者に簡素化されたスレッド・プログラミング環境を提供します。

革新的なスレッド化アシスタント

インテル® Parallel Advisor 2011: スレッド・アプリケーション設計を明確化および迅速化

コンパイラーとスレッド化ライブラリー

インテル® Parallel Composer 2011: C/C++ コンパイラーと高度なスレッド化ライブラリーで効率的なアプリケーションを開発

メモリーエラーとスレッド化のエラーチェッカー

インテル® Parallel Inspector 2011: 並列メモリーエラー / スレッドエラーを未然に防ぐ検証機能でアプリケーションの信頼性を確保

スレッドとパフォーマンス・プロファイラー

インテル® Parallel Amplifier 2011: スケーラブルなマルチコア・パフォーマンスが得られるようにボトルネックを迅速に検出してスレッド・アプリケーションをチューニング

