あなたのアプリケーションはコア数が増えると
さらに速くなるでしょうか?

アプリケーションのパフォーマンス・スケーラビリティーを測定する 2 つのツール

ホワイトペーパー

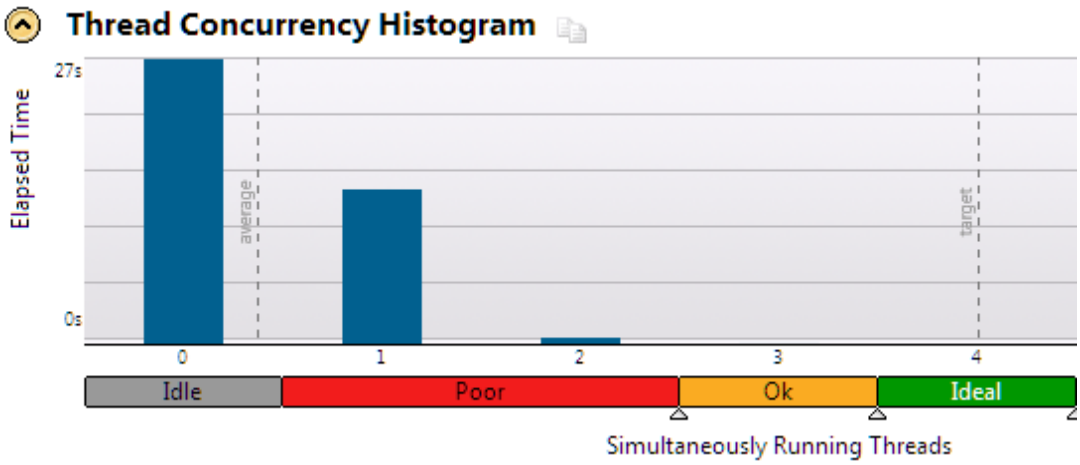
Q: 私のアプリケーションのパフォーマンスは、より多くのコア上で実行した場合にスケールしますか? より多くのコアを活用できるようにコードが十分並列化されているかどうか、どうやって確認できますか?

アプリケーションのコンカレンシー (並行性) を測定してみましょう

推測ではなく、正確に測定することが重要です。測定は、パフォーマンスを設計する上で鍵となります。アプリケーションは常に期待どおりに動作するとは限りません。アプリケーションが実際に何を行っているかを正確に把握することは、スケーラビリティーを効率良く得るために必要です。

インテル® VTune™ Amplifier XE (インテル® Parallel Studio XE に同梱) には複数のプロファイリング・ツールが備わっています。その中の 2 つのツールを実行するだけで、アプリケーションが現在のマルチコア・システムでどのように動作しているかが正確に判明し、またスケーリング・ボトルネックも分かります。

1 つ目のツール、「コンカレンシー解析」は、インテル® VTune™ Amplifier XE の主要なビューの 1 つです。このツールは、「特定のワークロードで、アプリケーションがシリアルおよび並列に実行している時間は?」という問いに答えます。これにより、ベストケースのパフォーマンス・スケーリングの指標が得られます。並列コードはスケーリングが可能ですが、シリアルコードはスケーリングしません。コンカレンシー解析により、並列コードがすべて完全にスケーリングした場合に達成可能な最大のパフォーマンスが分かります。

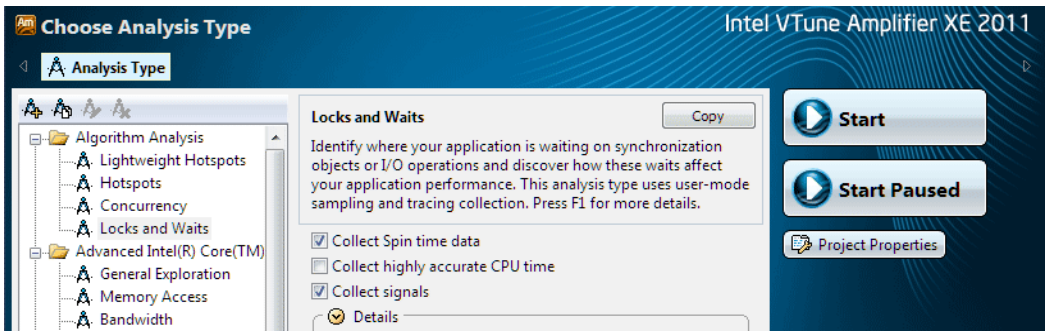


インテル® VTune™ Amplifier XE コンカレンシー・サマリー

上記は、4 コア・プロセッサーで実行されたアプリケーションの経過時間の解析結果で、特定の数のスレッドが同時に実行されているウォールクロック時間を示しています。スレッドが実際にプロセッサー上で実行しているか、OS スケジューラーで実行可能な状態であれば、スレッドは実行中と見なされます。

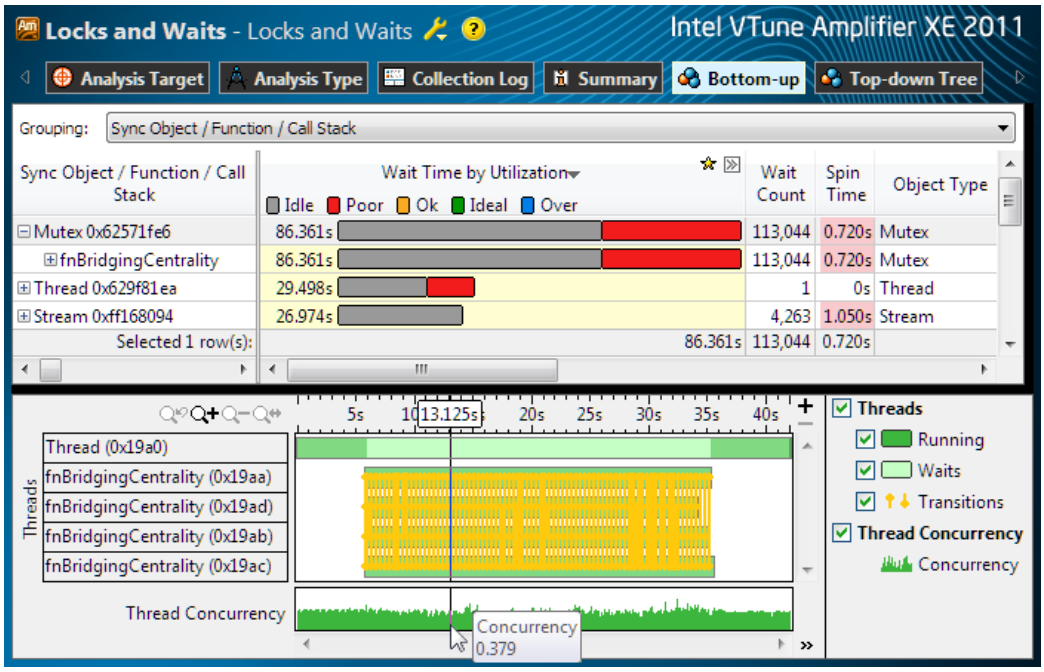
このアプリケーションは、適切にスケーリングしません。次の「ロックと待機の解析」でその理由が分かります。

2 つ目のツール、「ロックと待機の解析」では、スレッドが良く連携して動作しているかどうかを解析します。競合がたくさん発生している場合は、減らすべきでしょう。それにより、現在のマルチコア・システムでパフォーマンスが改善するだけでなく、コア数が増えたときのパフォーマンス・スケーラビリティーも向上します。



簡単な解析実行

インテル® VTune™ Amplifier XE でロックと待機の解析を実行するには、左側の解析タイプで [Locks and Waits] を選択し、[Start] ボタンをクリックするだけです。



ロックと待機の解析

上記の解析結果により、問題が明確に分かります。ほとんどの時間で 4 つのスレッドが実行されていますが、2 つの問題が見られます。1 つは、I/O 時にロック (Stream) が保持されているため、26.97 秒のアイドル時間があることです。もう 1 つは、同期のオーバーヘッドが非常に大きいことです。

上部のグリッドは、待機時間でソートされた同期オブジェクトのリストです。待機中のプロセッサー使用率が色別に表示されています。待機中にプロセッサーが十分に利用されていれば (緑)、待機時間が長くても問題ありません。この例では、プロセッサー使用率が低い (赤) かアイドル状態 (灰色) になっています。グリッドの一番下の行 (Stream) は、I/O 時のアイドル時間を示しています。

効率良いスケーラブルなコードを見つけるには、オーベーヘッドと待機時間を確認することが重要です。また、スレッドが数個のロックで長い時間待機しているか、または多くのロックで少しずつ待機しているのかを把握することも大切です。同期を示す下部のタイムラインでは、実行中のスレッド間の遷移が多発していることが分かります (黄色)。つまり、実際のコンカレンシー (タイムラインの下の行) は非常に低いことになります。遷移の多発は、過剰な同期の呼び出しである場合が多いので、コードやアルゴリズムを見直して同期の数や問題を減らす方法を検討すると良いでしょう。

アプリケーションはスケーリングするでしょうか?

以上で、アプリケーションがどのように動作しているかが判明しました。次に、基本的な設計の問題点を評価してみましょう。

スレッド数はハードウェアに合わせて調整されますか?

ハードウェアに合わせて自動的または簡単に最適なスレッド数に調整されるよう並列処理が設計されていますか? スレッドの数が少なすぎるとハードウェアの利点を活用できず、多すぎると非効率的です。インテル® スレッディング・ビルディング・ブロック (インテル® TBB) のようなハイレベルの並列化モデルであれば、これは自動で行われます。OS レベルのスレッド化を行っている場合、開発者が自身でこれを管理しなければなりません。

多くのコア数で最適な粒度になりますか?

各スレッドのワークロードは、スレッドのオーバーヘッドを打ち消すのに十分な大きさでなければなりません。インテル® VTune™ Amplifier XE のコンカレンシー解析を使用してこれを測定するか、あるいはプログラムの実行時間を計測します。仮に、測定した結果が、16 コア・システムで最適なワークロードであると示したとします。(適切なスケーリングの期待値は、プロジェクトにより異なります。) アプリケーションは 16 コアで最適に実行されますが、果たして 64 コア・システムではどうなるでしょうか? これを確認するには、現在のシステムでアプリケーションを再度解析します。ただし、より多くのコア数を持つ将来のシステム上のコアごとのワークロードのサイズに合わせて、ワークロードのサイズを変更します。これを行っても、スレッド化のオーバーヘッドが合理的であれば、粒度は将来のシステムでも問題ないでしょう。

同期のオーバーヘッドは増大しますか?

スレッド数が増えると、同期ポイントの数も増えますか? 増える場合、どの程度になりますか? 同期ポイント数が増えない場合、またはその増加がスレッド数に比例する場合、あるいは相互作用の程度が低く、グローバル (すべてのスレッド) ではない場合、その同期のオーバーヘッドは問題ないでしょう。不明な場合は、インテル® VTune™ Amplifier XE を使ってより多くのスレッドを実行し、同期の状態を確認してください。

このような設計問題を確認するときが、最適な並列化モデルを考える絶好の機会です。スケーラビリティーの設計で最も簡単な方法は、より高レベルな並列化モデルを採用することです。

並列処理をより高レベルで表現

アプリケーションをアセンブリーではなく、高水準言語で記述する生産性のメリットはご存じでしょう。並列処理も同様です。Pthreads や Windows* スレッドなどの低レベルな構造を使用することは、アセンブリーを記述するようなものです。これらは最大限の制御が可能ですが、生産性が低く、パフォーマンスを最適化するには新しいハードウェアで変更を加えなければなりません。

すべてのアプリケーションに適した高レベルの並列処理構造はありません。それぞれのニーズに応じて、多数の高レベル構造があります。

OpenMP* は、C および Fortran 開発者によく使用されており、インテルを含む多くの企業でサポートされています。OpenMP* がすでに最適に動作していれば、変更する理由はないでしょう。ただし、OpenMP* を使用しないプログラムには、より良い並列化のオプションがあります。OpenMP* は互換性に欠け、特に並列処理の入れ子をサポートしていません。インテル® TBB とインテル® Cilk™ Plus ではこれが可能です。それでも、すでに OpenMP* に多くの投資を行っているのであれば、OpenMP* は適切なソリューションといえます。

インテル® スレッディング・ビルディング・ブロック (インテル® TBB) は、C++ テンプレート・ライブラリー・ベースのソリューションです。タスクベースの抽象化を使用することで、より簡単にスケーラブルで信頼性の高い並列アプリケーションを記述できます。並列化モデルとしてインテル® TBB を使用すると、新しい世代のハードウェア向けにも最適化される、既知のスケーラブルなソリューションを利用できます。インテル® TBB の商用版には、Windows* 版、Linux* 版、Mac OS* 版があり、またオープンソース版もあります。インテル® TBB は幅広いアルゴリズムを表現することが可能です。スケーラブルなメモリー・アロケーターに加えて、高度に最適化されたスレッド制御を上級ユーザーに提供します。

インテル® Cilk™ Plus は、C/C++ の言語拡張です。インテル® Cilk™ Plus の 3 つのキーワード は、単純でありながら非常に強力な並列プロミング・モデルを提供します。また、ランタイムおよびテンプレート・ライブラリーは、きめ細やかにチューニングされた並列アプリケーションを作成する環境を提供します。インテル® Cilk™ Plus はインテル® Parallel Studio XE に含まれ、Cilk Plus のサブセットは、GCC 4.7 プロジェクトの開発ブランチに追加されました。インテル® Cilk™ Plus は非常に効率的で、コードの変更があまり必要ないため、ポピュラーな選択肢です。

Fortran 2008 規格で定義された Co-Array により、Fortran 開発者は並列プログラミングが可能になりました。Fortran 言語拡張である Co-Array は、堅固で効率的な並列プログラミング言語として Fortran を使用する 1 つの手法を提供します。Co-Array Fortran は SPMD (Single Program Multiple Data) プログラミング・モデルを使用し、インテル® Parallel Studio XE でサポートされています。

インテル® Parallel Studio XE で利用できる並列プログラム開発の選択肢の例をいくつか紹介しましたが、ご自身のアプリケーションに最適なものを選んでください。どのオプションを選択しても、より高レベルの抽象化へ移行するメリットを実感できるでしょう。生産性が向上し、高レベルの並列化を実装するコンパイラーやライブラリーによって、新しいハードウェア・プラットフォームでのパフォーマンスが最適化されます。

まとめ

マルチコア/メニーコアが普及する中で、コンカレンシーとスケーラビリティーは、将来のパフォーマンス向上には不可欠です。ツールを使ってアプリケーションのパフォーマンスを特定することは、より多くのコア数を持つ将来のハードウェアにおけるパフォーマンスを予測する重要な鍵となります。高レベルの抽象概念で並列処理を表現することは、生産性を向上し、パフォーマンスのスケーラビリティーの可能性を高めます。

評価版

インテル® Parallel Studio XE の評価版をダウンロードしてお試しください。次のコンポーネントが含まれています。

- インテル® VTune™ Amplifier XE パフォーマンス・アナライザー
- 並列プログラミング・モデル (インテル® TBB) も含まれます。
- インテル® Inspector XE メモリー/スレッド チェッカー
- スタティック・セキュリティー解析: コーディング・エラーを発見し、ソフトウェアのセキュリティーを強化します。
- 最適化コンパイラーとパフォーマンス・ライブラリー

推奨するドキュメント

- マルチスレッド・アプリケーション開発のためのガイド
- Web セミナー – 「The Key to Scaling Applications for Multicore (マルチコア向けにアプリケーションをスケーリングするための鍵)」
- インテル® VTune™ Amplifier XE – 概要ビデオ と 詳細な手順の入門チュートリアル
- インテル® TBB ホワイトペーパー:
  - インテル® TBB のコンカレント・コンテナーによる安全でスケーラブルな並列処理
  - インテル® TBB の汎用並列アルゴリズムによるスケーラブルな並列処理の解説
  - インテル® TBB: マルチコア向けのスケーラブルなプログラミング

製品情報および購入情報は、インテル® ソフトウェア開発製品 Web サイトを参照してください。
http://www.intel.co.jp/jp/software/products/

著者について

Dick Kaiser は、インテル コーポレーションのプロダクト・マーケティング・エンジニアで、ソフトウェア解析ツールを担当しています。

著作権と商標について

JPN/1204/PDF/XL/SSG/KS Intel_TwoToolsMeasurethePerformanceScalabilityofYourApplication_WP /Rev1011