

スレッドについての考察

この記事は、CodeProject* に公開されている「Thoughts About Threads」の日本語参考訳です。

TotalView* は、科学および学術研究分野の開発者に、プロセスとスレッド実行の制御、そしてプログラムの状態とデータに関する詳細な情報を提供するツールセットです。

編集者注

この記事は、CodeProject* の features > Product Showcase セクションにあります。Product Showcase セクションでは、開発者に役立つ CodeProject* スポンサーの製品とサービスに関する情報を紹介しています。

[インテル® コンパイラーを入手する \(英語\)](#)

数十年にわたり 2 年ごとにプロセッサのパフォーマンスが倍増することを確実に予測してきたムーアの法則は、新たな課題を示しています。従来のプロセッサは、量子物理学レベルと高速化により生じる熱量への対処において限界に直面しています。このため、インテルや AMD* などの半導体メーカーは、マルチコア・テクノロジーと並列化によりプロセッサ能力の向上を図っています。

ソフトウェア・エンジニアが複雑なマルチスレッド・コードやマルチコアコードの作成に関心を持ち始めたのは最近のことです。それまでは、シングルスレッド・コードを記述して、ハードウェア・ベンダーが 2 ~ 3 年ごとに発表する新しいアーキテクチャーを利用することでスピードアップを実現してきました。ソフトウェア・エンジニアが何もしなくても、アプリケーションをより高速に実行できていたのです。

並列化への関心の高まり

インテル® Xeon Phi™ 製品ファミリーなどの新しく強力なアーキテクチャーの登場によってすべてが変わりました。追加のコアを利用するため、半導体メーカーはハードウェアとソフトウェアの両方を並列化する必要がありました。それによってもたらされた利点の 1 つが、レイテンシーを下げ、スループットを上げることで可能になった速度の向上です。これは、単一のジョブ (例えば、電話交換アプリケーション) のウォールクロック時間の短縮に取り組んでいる開発者や、特定の時間内にほとんどのジョブを処理できるようにスループットの改善に取り組んでいる開発者にとって特に重要です。

開発者にとってこの変化は、設計段階の初期からアプリケーションを並列実行する方法を考える必要があることを意味します。幸いにも、マルチスレッドを利用したり、複数のプロセッサを並行して実行するなど、いくつかの選択肢があります。マルチコアシステムでは、特定のアプリケーションに専用のプロセッサを使用することがあります。例えば、購買発注処理システムでは、専用のプロセッサまたはスレッドで Word* ベースのアプリケーションを実行し、Excel* 表計算は別のプロセッサ/スレッドで処理して、Internet Explorer* の処理はさらに別のプロセッサ/スレッドが行うことで、これらのアプリケーションをすべて並列に実行できます。その間にプロセス間通信も行われる可能性があります。

システムがサポートするマルチスレッド機能を利用する並列コードの開発は複雑です。マルチスレッド・システム向けアプリケーションの作成方法を学ぶことは、ソフトウェア・プログラミングにおいて最も困難なトピックの1つです。以前は、プログラマーはシングルスレッドのみを扱っていました。マルチスレッド処理向けに設計された複雑なアプリケーションでは、一部を変更するとシステム全体に影響し、すべての変更が必要となり、複雑さが増します。

スレッドの整合性と安全性も問題になります。新しいスレッドが既存のスレッドと適切に同期されない可能性もあります。その場合、問題の診断が非常に困難です。たった1ミリ秒の整合性のない動作により、複数の問題が別々に発生しているように見えます。

インテルと Rogue Wave* によるソフトウェアのコラボレーション

デバッグは、マルチスレッド環境において最も困難なタスクであり、開発者は新たな問題に自力で取り組むか、あるいはツールを利用することで複雑さを大幅に軽減し、抽象化することができます。それらのツールは、C/C++ や古くからある FORTRAN などのさまざまな言語に対応しています。インテル® VTune™ Amplifier や HPC 向けの TotalView (Rogue Wave* のデバッガー) はその一例です。インテルと Rogue Wave* の最新のコラボレーションにより、TotalView Xeon Phi Debugger が誕生しました。

Rogue Wave* によると、TotalView* は数百から数千のコアで実行される大規模並列アプリケーションとマルチコア・アプリケーションの迅速な障害分離、メモリー最適化の改善、動的な視覚化をサポートしています。TotalView* は、科学および学術研究分野の開発者に、プロセスとスレッド実行の制御、そしてプログラムの状態とデータに関する詳細な情報を提供するツールセットです。

Rogue Wave* の主任製品マネージャーである Marty Bakal 氏は、「TotalView* は、単一のウィンドウで**多数のプロセスとスレッド**を同時にデバッグすることが可能です。開発者は、プログラムの実行を完全に制御できます。」と述べています。このため、開発者は、単一のスレッドまたは任意のプロセッサ/スレッドのグループでアプリケーションを実行、ステップ実行、および停止できます。また、リバースデバッグにより障害の個所から逆方向に巻き戻して作業できるため、アプリケーションを繰り返し再起動しなくても素早く原因を切り分け、スレッド、OpenMP*、MPI、GPU、コプロセッサを利用する並列プログラムで発生する困難な問題を再現してトラブルシューティングできます。

TotalView* は、開発者がインテル® Xeon Phi™ プロセッサ上で表示、制御、デバッグできるように完全なサポートを提供します。例えば、TotalView* には次の機能があります。

- ネイティブ、オフロード、対称プログラミング・モデルをサポート
- インテル® Xeon Phi™ プロセッサ上のネイティブ・アプリケーションと対称アプリケーションでメモリーのデバッグをサポート
- 完全な非同期スレッド制御
- 特定のブレークポイントの共有
- クラスタとマルチデバイス構成をサポート
- インテル® Xeon Phi™ プロセッサで MPI と MPI + OpenMP* ハイブリッドのネイティブ起動をサポート

今後の展開

今日、インテルは、主に 24 コア (6 CPU、それぞれ 4 コア) 搭載のマシンを市場に提供しています。しかし、今後数年間のプロセッサの進化を予測したロードマップでは、100 コアを搭載したマシンの登場が予測されています。そのためには、いくつかの主要な速度上の課題に対応する必要があります。例えば、従来のソフトウェアを効率良く並列に実行できるようにチューニングし、システム全体をデバッグして、8 コアのマシンですべてのコアを利用して実行できるようになったとします。しかし、このソフトウェアを 96 コアのマシンに移植すると、すべてが台無しになります。この問題は解決できますが、I/O とデータアクセスの複雑さによって生じる新たなボトルネックなどさまざまな課題に直面するでしょう。

以下は、近い将来に生じる可能性があるいくつかのシナリオとヒントです。

- すべてのラックがバックプレーンを共有し 1 つの大きな箱のように見える、高速バックプレーンを搭載したシステムが設計されるでしょう。アプリケーションは、128 コアが利用可能な 1 台のマシンとして動作します。
- 将来のソフトウェア・ソリューションは、MPI や OpenMP* などの既存のアプリケーションを使用するでしょう。OpenMP* はスレッド化をコンパイラーに任せ、開発者のタスクを軽減します。
- ヒント – 並列化について開発サイクルの早期に考慮して、その力と複雑さを以降の設計段階で最大限に活用します。
- すべての利用可能なツールについても開発段階の早期 (できれば設計段階の初期) に調査します。
- 最後に、スレッド化とマルチプロセッシングを含む並列プログラミングの基礎知識を得ることが重要です。全く新しいソフトウェア開発の世界が広がっており、並列処理はその中心と言えます。

この記事は、最先端の技術を通じて HPC コミュニティーが推進する最新の科学、研究、イノベーションを紹介するため、インテルの HPC エディトリアル・プログラムの一環として作成されました。最終的な編集権と記事発行の決定権はコンテンツの発行者が有します。

ライセンス

この記事と関連するソースコードおよびファイルは、[The Code Project Open License \(CPOL\)](#) (英語) の下に利用許諾されます。