インテル® Xeon® スケーラブル・プロセッサー向け SIMD ベクトル化のチューニング

この記事は、インテル® デベロッパー・ゾーンに公開されている「Tuning SIMD vectorization when targeting Intel® Xeon® Processor Scalable Family」の日本語参考訳です。

はじめに

インテル® Xeon® スケーラブル・プロセッサーは、Skylake† サーバー・マイクロアーキテクチャーをベースにしています。

インテル® Xeon® スケーラブル・プロセッサーで最高のパフォーマンスを引き出すには、インテル® C++ および Fortran コンパイラーではプロセッサー固有の [Q]xCORE-AVX512 オプションを指定してコンパイルすべきです。このオプションでビルドされたアプリケーションは、インテル製以外のプロセッサーまたはインテル® AVX-512 命令セットをサポートしないプロセッサーでは動作しません。

アプリケーションは、複数のプロセッサー向けに複数の命令セットを利用してコンパイルすることもできます。例えば、[Q]axCORE-AVX512,CORE-AVX2 を指定すると、CORE-AVX512 (開発コード名 Skylake Server) と CORE-AVX2 (開発コード名 Haswell または Broadwell) のそれぞれのプロセッサー向けに最適化されたコードパスに加えて、デフォルトのインテル® ストリーミング SIMD 拡張命令 2 (インテル® SSE2) コードパスを含む fat バイナリーが生成されます。インテル® Xeon® スケーラブル・プロセッサーとインテル® Xeon Phi™ x200 製品ファミリー向けの共通バイナリーを生成するには、[Q]xCOMMON-AVX512 オプションを指定してコンパイルします。

変更点

インテル® Xeon® スケーラブル・プロセッサーで最も広い 512 ビット・ベクトル幅を選択しても、すべてのループ、特に HPC 以外のアプリケーションでよく見られるトリップカウントが少ないループでは、必ずしも最適なベクトル化されたコードが生成されるわけではありません。

いくつかの分野のアプリケーションを慎重に検討した結果、インテル® Xeon® スケーラブル・プロセッサーの SIMD ベクトル化に柔軟性を持たせるため、デフォルトでは 512 ビット ZMM の使用率を低く設定し、利点が得られる場合は高く設定できるようにしました。インテル® コンパイラーの最適化レポートまたはインテル® Advisor を利用することで、SIMD ベクトル化の効率を理解し、さらなる可能性を探ることができます。

インテル® コンパイラー 18.0 および 17.0.5 以降では、256 ビット・ベクトル・レジスターを利用するインテル® アドバンスト・ベクトル・エクステンション 2 (インテル® AVX2) から 512 ビット・ベクトル・レジスターを利用するインテル® アドバンスト・ベクトル・エクステンション 512 (インテル® AVX-512) へのスムーズな移行を可能にするため、新しいコンパイラー・オプション [Q/q]opt-zmm-usage=low|high が追加されました。この新しいオプションは、[Qa]xCORE-AVX512 オプションとともに使用します。

[Qa]xCORE-AVX512 のデフォルトでは、インテル® コンパイラーにより ZMM レジスターの使用が限定されます。これにより、ある種のアプリケーションは最適に動作します。そうでないアプリケーション (例えば、多くの浮動小数点演算を含むアプリケーション) では、ZMM レジスターを使用して積極的に 512 ビット SIMD ベクトル化を行う新しいオプション [Q/q]opt-zmm-usage=high により利点が得られる可能性があります。

512 ビット SIMD ベクトル化で ZMM レジスターの使用率を高めるためにすべきこと

この目的を達成する方法は 3 つあります。以下は、説明を目的としたサンプルコードです。

```
$ cat Loop.cpp
#include <math.h>
void work(double *a, double *b, int size)
{
    #pragma omp simd
    for (int i=0; i < size; i++)
    {
        b[i]=exp(a[i]);
    }
}
```

方法 1: インテル® コンパイラー 18.0 および 17.0.5 以降では、新しいコンパイラー・オプション [Q/q]opt-zmm-usage=high を [Qa]xCORE-AVX512 とともに指定することで、ZMM レジスターの使用率が高くなり、完全な 512 ビット SIMD ベクトル化が行われる可能性があります。この新しいオプションを使用するためにソースコードを変更する必要はありません。オプションを指定するだけで、簡単にコンパイル単位全体で ZMM の使用率を高めることができます。

デフォルトのオプションでコンパイルすると、コンパイラーは新しいオプションの使用を推奨するリマークを出力します。

```
$ icpc -c -xCORE-AVX512 -qopenmp -qopt-report:5 Loop.cpp
…
リマーク #15305: ベクトル化のサポート: ベクトル長 4
…
リマーク #15321: コンパイラーは XMM/YMM ベクトルをターゲットとして選択しました。ZMM を使用するには、-qopt-zmm-usage=high を指定してオーバーライドしてください。
…
リマーク #15476: スカラーのコスト: 107
リマーク #15477: ベクトルのコスト: 19.500
リマーク #15478: スピードアップの期待値: 5.260
…
```

推奨された新しいオプションを追加してコンパイルすると、上記のリマークは出力されなくなり、ZMM の使用率が高くなることで SIMD によるパフォーマンス・ゲインが向上し、コードがスピードアップします。

```
$ icpc -c -xCORE-AVX512 -qopt-zmm-usage=high -qopenmp -qopt-report:5 Loop.cpp
…
リマーク #15305: ベクトル化のサポート: ベクトル長 8
…
リマーク #15476: スカラーのコスト: 107
リマーク #15477: ベクトルのコスト: 9.870
リマーク #15478: スピードアップの期待値: 10.110
…
```

方法 2: 新しいコンパイラー・オプションを使用する代わりに、アプリケーションで OpenMP* simd 構文の simdlen 節を使用してより広いベクトルを指定し、512 ビット・ベースの SIMD ベクトル化を達成することができます。この変更は、特定のループに対して行われるため、ホットスポットのチューニングの慣例に従って、該当するループごとに適用する必要があります。そのため、この方法ではある程度のコード変更が必要になります。

サンプルコードでは、simdlen 節を使用することで SIMD によるパフォーマンス・ゲインが向上します。ここで、simdlen(8) は、8 つの要素がオーバーラップしないことを示します。

```
#pragma omp simd simdlen(8)
for (int i=0; i < size; i++) …

$ icpc -c -xCORE-AVX512 -qopenmp -qopt-report:5 Loop.cpp
…
リマーク #15305: ベクトル化のサポート: ベクトル長 8
…
リマーク #15476: スカラーのコスト: 107
リマーク #15477: ベクトルのコスト: 9.870
リマーク #15478: スピードアップの期待値: 10.110
…
```

方法 3: [Qa]xCOMMON-AVX512 オプションでビルドされたアプリケーションは、すでに ZMM レジスターの使用率が高いため、上記の 2 つの方法を試す必要はありません。ただし、このアプリケーションは、インテル® Xeon® スケーラブル・プロセッサーやインテル® Xeon Phi™ x200 製品ファミリーのようなインテル® AVX-512 の共通セットをサポートするプロセッサーでは動作しますが、[Qa]xCOMMON-AVX512 で生成されないインテル® AVX-512 命令のサブセットをサポートするプロセッサーでは動作しません。アプリケーションの種類によっては、デフォルトの [Q/q]opt-zmm-usage=low オプションを指定したほうがパフォーマンスが良い可能性があります。

まとめ

インテル® Xeon® スケーラブル・プロセッサー向けに計算負荷の高いアプリケーションをビルドする場合、ここで述べた方法で ZMM レジスターの使用率を高めて積極的に 512 ビット SIMD ベクトル化を利用することができます。

†開発コード名

コンパイラーの最適化に関する詳細は、最適化に関する注意事項を参照してください。