Intel コンパイラ 利用の手引き

TSUBAME Computing Services, Center for Information Infrastructure

2025-01-31

Table of contents

1. はじめに 3
1.1. 利用できるバージョン 3
1.2. マニュアル 3
2. 利用方法 4
2.1. Intelコンパイラ 4
2.2. 並列化 6
2.3. 数値計算ライブラリ 7

1. はじめに

Info

本ページのコマンドライン例では、以下の表記を使用します。
[login]$: ログインノード
[rNnN]$: 計算ノード
[login/rNnN]$: ログインノードまたは計算ノード
[yourPC]$: ログインノードへの接続元環境

本書は、Intelコンパイラ をTSUBAME4.0 で利用する方法について説明しています。
また、TSUBAME4を利用するにあたっては、TSUBAME利用の手引きもご覧下さい。 サーバの利用環境や注意事項などが詳細に記述されていますので、よく読んでください。

1.1. 利用できるバージョン

TSUBAME4で利用可能な最新バージョンについてはTSUBAME計算サービスWebサイトの アプリケーション ページをご確認下さい。
研究に支障がない限り、バグ修正の入っている最新版をご利用下さい。

1.1.1. バージョンの切り替え

本システムでは、moduleコマンドを使用することでコンパイラやアプリケーション利用環境の切り替えを行うことができます。 特にバージョンの指定がない場合は、 バージョン 2024.0.2 がロードされるようになっています。

例: Ver 2024.0.2

```
[login/rNnN]$ module load intel/2024.0.2
```

1.2. マニュアル

マニュアルはTSUBAMEの次のディレクトリ下にあります。

バージョン	ディレクトリ
2024.0.2	{PATHを要修正}/en {PATHを要修正}/ja

オンラインドキュメントも公開されています。

Intel® oneAPI DPC++/C++ Compiler
Intel® Fortran Compiler
Intel® VTune™ Profiler
Intel® Trace Analyzer and Collector
Intel® Inspector
Intel® oneAPI Math Kernel Library (oneMKL)
Intel® MPI Library

2. 利用方法

2.1. Intelコンパイラ

Intelコンパイラの各コマンドは以下のとおりです。

コマンド名とコマンド形式

コマンド	言語	コマンド形式
ifx	Fortran 77/90/95/2003/2008/2018	`$ ifx [オプション] source_file`
icx	C	`$ icx [オプション] source_file`
icpx	C++	`$ icpx [オプション] source_file`

利用する際は、moduleコマンドでintelを読み込んでください。 --helpオプションを指定して頂くとコンパイラオプションの一覧が表示されます。

Info

Intel oneAPI 2024から、iccコマンドおよびicpcコマンドが使用できなくなりました。 icxコマンドおよびicpxコマンドを使用してください。
また、Intel oneAPI 2025から、ifortコマンドが使用できなくなりました。
ifxコマンドを使用してください。

2.1.1. コンパイルの主なオプション

コンパイルの最適化オプションを以下に示します。

コンパイラの主なオプション

オプション	説明
`-O`	O2 と同じ。
`-O0`	すべての最適化を無効にします。
`-O1`	最適化を有効にします。コードサイズを大きくするだけで高速化に影響を与えるような一部の最適化を無効にします。
`-O2`	最適化を有効にします。一般的に推奨される最適化レベルです。 ベクトル化は O2 以上のレベルで有効になります。-Oオプションを指定しない場合、デフォルトでこちらが指定されます。
`-O3`	O2 よりも積極的に最適化を行い、融合、アンロールとジャムのブロック、IF 文の折りたたみなど、より強力なループ変換を有効にします。
`-xCORE-AVX512`	Intel プロセッサー向けのIntel アドバンスト・ベクトル・エクステンション 512 (Intel AVX512)、Intel AVX2、AVX、SSE4.2、SSE4.1、SSSE3、SSE3、SSE2、SSE 命令を生成します。Intel AVX2 命令セット対応のIntel プロセッサー向けに最適化します。
`-xCORE-AVX2`	Intel プロセッサー向けのIntel アドバンスト・ベクトル・エクステンション 2 (Intel AVX2)、Intel AVX、SSE4.2、SSE4.1、SSSE3、SSE3、SSE2、SSE 命令を生成します。Intel AVX2 命令セット対応のIntel プロセッサー向けに最適化します。
`-xSSE4.2`	Intel プロセッサー向けのIntel SSE4 高効率および高速な文字列処理命令、Intel SSE4 ベクトル化コンパイラ命令およびメディア・アクセラレーター命令、およびIntel SSSE3、SSE3、SSE2、SSE 命令を生成します。Intel SSE4.2 命令セット対応のIntel プロセッサー向けに最適化します。
`-xSSSE3`	Intel プロセッサー向けのIntel SSSE3、SSE3、SSE2、SSE 命令を生成します。Intel SSSE3 命令セット対応のIntel プロセッサー向けに最適化します。xオプションを指定しない場合、デフォルトでこちらが指定されます。
`-qopt-report=n`	最適化レポートを生成します。デフォルトでは、レポートは.optrpt 拡張子を持つファイルに出力されます。　には、0 (レポートなし) から5 (最も詳しい) の詳細レベルを指定します。デフォルトは 2 です。
`-fp-model precise`	浮動小数点演算のセマンティクスを制御します。浮動小数点データの精度に影響する最適化を無効にし、中間結果をソースで定義された精度まで丸めます。
`-g`	-gオプションはオブジェクト・ファイルのサイズを大きくするシンボリック・デバッグ情報をオブジェクト・ファイルに生成するようにコンパイラに指示します。
`-traceback`	このオプションは、ランタイム時に致命的なエラーが発生したとき、ソースファイルのトレースバック情報を表示できるように、オブジェクト・ファイル内に補足情報を生成するようにコンパイラに指示します。 致命的なエラーが発生すると、コールスタックの 16 進アドレス (プログラム・カウンター・トレース) とともに、ソースファイル、ルーチン名、および行番号の相関情報が表示されます。 マップファイルとエラーが発生したときに表示されるスタックの 16 進アドレスを使用することで、エラーの原因を特定できます。 このオプションを指定すると、実行プログラムのサイズが増えます。

2.1.1.1. コンパイルの推奨最適化オプション

コンパイルの推奨最適化オプションを以下に示します。本システムに搭載している AMD EPYC 9654 は、Intel AVX512 命令セットに対応していますので、-xCORE-AVX512オプションを指定することができます。-xCORE-AVX512を指定すると、コンパイラがソースコードを解析し、最適なAVX512、AVX2、AVX、SSE命令を生成します。推奨最適化オプションは積極的な最適化を行い、かつ安全なオプションです。最適化のために計算の順序を変更する可能性があり、結果に誤差が生じる場合があります。

> **Info**
>
> Intel Xeon (Skylake) 以降で採用された AVX-512 は、本システムに搭載している Zen4 アーキテクチャの AMD 第4世代 EPYC で対応しました。

オプション	説明
`-O3`	O2 最適化を行い、融合、アンロールとジャムのブロック、IF 文の折りたたみなど、より強力なループ変換を有効にします。
`-xCORE-AVX512`	Intel プロセッサー向けのIntel アドバンスト・ベクトル・エクステンション 512 (Intel AVX512)、Intel AVX2、AVX、SSE4.2、SSE4.1、SSSE3、SSE3、SSE2、SSE 命令を生成します。Intel AVX2 命令セット対応のIntel プロセッサー向けに最適化します。

上記のオプションを使用することにより、プログラムの性能が悪化した場合、最適化のレベルを-O2に下げるかベクトル化のオプションを変更してください。 また、結果が一致していない場合、浮動小数点のオプションも試してみてください。

2.1.2. Intel 64アーキテクチャーのメモリモデル指定

次のいずれかのメモリモデルを使用して実行バイナリを作成します。

メモリモデル

メモリモデル	説明
small (-mcmodel=small)	コードとデータのすべてのアクセスが、命令ポインター (IP) 相対アドレス指定で行われるように、コードとデータはアドレス空間の最初の 2GB までに制限されます。 -mcmodelオプションを指定しない場合、デフォルトでこちらが指定されます。
medium (-mcmodel=medium)	コードはアドレス空間の最初の 2GB までに制限されますが、データは制限されません。コードは IP 相対アドレス指定でアクセスできますが、データのアクセスは絶対アドレス指定を使用する必要があります。
large (-mcmodel=large)	コードもデータも制限されません。コードもデータもアクセスは絶対アドレス指定を使用します。

IP 相対アドレス指定は 32 ビットのみ必要ですが、絶対アドレス指定は 64 ビット必要です。これは、コードサイズとパフォーマンスに影響します。(IP 相対アドレス指定の方が多少速くアクセスできます。)
プログラム内の共通ブロック、グローバルデータ、静的データの合計が2GBを越えるとき、リンク時に次のエラーメッセージが出力されます。

```
<some lib.a library>(some .o): In Function <function>:
  : relocation truncated to fit: R_X86_64_PC32 <some symbol>
.....
  : relocation truncated to fit: R_X86_64_PC32 <some symbol>
```

この場合は、-mcmodel=mediumと-shared-intelを指定してコンパイル/リンクして下さい。
medium メモリモデルまたは large メモリモデルを指定した場合、Intelのランタイム・ライブラリの適切なダイナミック・バージョンが使用されるように、-shared-intel コンパイラ・オプションも指定する必要があります。

2.2. 並列化

2.2.1. スレッド並列(OpenMP)

OpenMPによるスレッド並列によりプログラムの高速化ができます。
OpenMPを使用する場合のコマンド形式を以下に示します。

コマンド形式(OpenMP)

	言語	コマンド形式
OpenMP	Fortran 77/90/95/2003/2008/2018	`$ ifx -qopenmp [オプション] source_file`
	C	`$ icx -qopenmp [オプション] source_file`
	C++	`$ icpx -qopenmp [オプション] source_file`

'-opt-report-phase=openmp" オプションを使用することでOpenMP最適化フェーズのレポートを作成することができます。

2.2.2. プロセス並列(MPI)

Fortran/C/C++ プログラムに MPIライブラリをリンクし、プロセス並列プログラムを作成/実行することができます。MPIを使用する場合のコマンド形式を以下に示します。利用する際は、moduleコマンドで各MPIを読み込んでください。

MPIライブラリ	言語	コマンド形式
Intel MPI	Fortran 77/90/95/2003/2008/2018	`$ mpiifx [オプション] source_file`
	C	`$ mpiicx [オプション] source_file`
	C++	`$ mpiicpx [オプション] source_file`
Open MPI	Fortran 77/90/95/2003/2008/2018	`$ mpifort [オプション] source_file`
	C	`$ mpicc [オプション] source_file`
	C++	`$ mpicxx [オプション] source_file`

2.3. 数値計算ライブラリ

2.3.1. Intel MKL

Intelマス・カーネル・ライブラリ (Intel MKL) は、工学、科学、金融系アプリケーション向けに高度に最適化され、広範囲にスレッド化された数学関数を含むライブラリです。
Intel MKL は、線形代数、高速フーリエ変換 (FFT)、ベクトルマス、直接法および反復法スパースソルバー、乱数ジェネレーター、その他を含む包括的なサポートを提供します。

- BLAS
- BLACS
- LAPACK
- ScaLAPACK
- PBLAS
- スパースソルバー
- 拡張固有値ソルバー
- ベクトル数学関数 (VML)
- 統計関数 (VSL)
- コンベンショナル DFT とクラスタ DFT
- 偏微分方程式のサポート
- 非線形最適化問題ソルバー
- データ・フィッティング関数
- FFTWへのラッパー・インタフェース

2.3.2. Intel MKLのリンク方法

Intel MKLのリンク方法は用途に応じてオプションを選んでください。コンパイラ・オプション -qmklを利用してリンクすることができます。
FFTWへのラッパー・インタフェースを利用するとき、コンパイル時にFFTWのヘッダーファイルが必要です。コンパイル時に、`-I $MKLROOT/include/fftw` を指定して下さい。
スレッド並列化版のIntel MKL(Intel コンパイラ)を以下に示します。

-mklオプションを使用したリンク方法(Intel MKL逐次版)

オプション	説明
`-qmkl=sequential`	BLAS, LAPACK, DFT, FFTW(ラッパー・インタフェース)

-qmklオプションを使用したリンク方法(Intel MKLスレッド並列化版)

オプション	説明
`-qmkl=parallel`	BLAS, LAPACK, DFT, FFTW(ラッパー・インタフェース)

【補足】
- 以下のページでは適切なリンク行を自動的に生成します。
Intel® oneAPI Math Kernel Library Link Line Advisor
MKLの使用方法　ダイナミックリンクまたはスタティックリンク、逐次版またはスレッド並列化版など　を指定しますと適切なリンク行を表示します。

2.3.3. 64ビット整数型に対応したライブラリのリンク

64ビット整数型に対応したライブラリ　ILP64ライブラリ　は以下の目的で提供されています。

・大規模なデータ配列 (要素数 2^31-1 以上) をサポートする
・`-i8` コンパイラ・オプションを使用して Fortran コードをコンパイルできるようにする

上記で、`-qmkl` オプションが指定されていますが、ILP64ライブラリをリンクするときには、`-qmkl=ilp64` オプションを指定して下さい。

ILP64用のコンパイル
- Fortran : `ifx -i8 …`
- C : `icx -DMKL_ILP64 …` - C++ : `icpx -DMKL_ILP64 …`

【補足】
- `-i8` または `-DMKL_ILP64` オプションを使用してコンパイルしたプログラムと LP64 ライブラリをリンクすると、予測できない結果や誤出力が発生する場合があります。

2.3.4. 実行スレッド数

ジョブスクリプト内の `OMP_NUM_THREADS` で設定したスレッド数で実行されます。
ジョブスクリプトについては、SMP並列を参照してください。
MKLのスレッド数のみを変更したいときはMKLの環境変数 `MKL_NUM_THREADS` を設定してください。