インテル® C++ コンパイラー 19.0 Update 4 for Linux* リリースノート (インテル® Parallel Studio XE 2019 Update 4)

このドキュメントでは、新機能、変更された機能、注意事項、および製品ドキュメントに記述されていない既知の問題について説明します。

パッケージに含まれるライセンスと本リリースノートの「著作権と商標について」をお読みください。本リリースのインテル® C++ コンパイラー 19.0 についての詳細は、次のリンクを参照してください。

変更履歴

インテル® C++ コンパイラー 19.0.3 以降 (インテル® C++ コンパイラー 19.0.4 での変更)

- 報告された問題を修正

インテル® C++ コンパイラー 19.0.2 以降 (インテル® C++ コンパイラー 19.0.3 での変更)

- 報告された問題を修正

インテル® C++ コンパイラー 19.0.1 以降 (インテル® C++ コンパイラー 19.0.2 での変更)

- インテル® C++ コンパイラー 19.0 Update 2 には機能とセキュリティーに関する更新が含まれます。ユーザーは最新のバージョンに更新する必要があります。

インテル® C++ コンパイラー 19.0 以降 (インテル® C++ コンパイラー 19.0.1 での変更)

- #pragma omp simd 向けの精度に影響しない simd オプション
- -[Q]x / -[Q]ax / -[m]tune / -[m]arch オプションで新しい開発コード名をサポート
- カスタム・メモリー・アロケーター・ライブラリー

インテル® C++ コンパイラー 18.0 以降 (インテル® C++ コンパイラー 19.0 での変更)

- OpenMP* parallel プラグマのユーザー定義のインダクションをサポート
- 排他スキャン simd をサポート
- Ubuntu* 18.04 での glibc 問題の解決を含む問題の修正
- -qopenmp-simd をデフォルトで設定
- -rcd オプションを廃止
- cannonlake オプションをサポート
- 投機的実行のサイドチャネル問題を軽減するための変更と新しい -mindirect-branch オプション。詳細は、「インテル® コンパイラーを使用して投機実行サイドチャネルの問題を緩和」を参照してください。
- 新しい C++17 機能をサポート
- vector プラグマの nodynamic_align 節と vectorlength 節
- OpenMP* TR6 Version 5.0 Preview 2 の部分サポートを拡張
- 新規および変更されたコンパイラー・オプション

先頭へ戻る

動作環境

- RAM 2GB (4GB 推奨)
- 7.50GB のディスク空き容量 (すべての機能をインストールする場合)
- 次の Linux* ディストリビューションのいずれか (本リストは、インテル社により動作確認が行われたディストリビューションのリストです。その他のディストリビューションでも動作する可能性はありますが、推奨しません。ご質問は、テクニカルサポートまでお問い合わせください。)
    - Debian* 8.0、9.0
    - Fedora* 27、28
    - Red Hat* Enterprise Linux* 6、7
    - SUSE* Linux* Enterprise Server 12 (SP5)、15
    - Ubuntu* 18.04 LTS、17.10

    - CentOS* 7.1、7.2
    - インテル® Cluster Ready
- Linux* 開発ツール・コンポーネント (gcc、g++ および関連ツールを含む)
    - gcc バージョン 4.3 - 8.x をサポート
    - binutils バージョン 2.20-2.29 をサポート
- -traceback オプションを使用するには、libunwind.so が必要です。一部の Linux* ディストリビューションでは、別途入手して、インストールする必要があります。

Eclipse* 開発環境に統合するためのその他の条件

- Eclipse* Platform 4.7 および次の両方
    - Eclipse* C/C++ Development Tools (CDT) 9.2.x
    - Java* ランタイム環境 (JRE) 8.0 (1.8) 以降
- Eclipse* Platform 4.8 および次の両方
    - Eclipse* C/C++ Development Tools (CDT) 9.3.x
    - Java* ランタイム環境 (JRE) 8.0 (1.8) 以降

注

- インテル® コンパイラーは、さまざまな Linux* ディストリビューションと gcc バージョンで動作確認されています。一部の Linux* ディストリビューションには、動作確認されたヘッダーファイルとは異なるバージョンのものが含まれており、問題を引き起こすことがあります。使用する glibc のバージョンは、gcc のバージョンと同じでなければなりません。最良の結果を得るため、上記のディストリビューションで提供されている gcc バージョンのみを使用してください。
- 非常に大きなソースファイル (数千行以上) を -O3、-ipo および -openmp などの高度な最適化オプションを使用してコンパイルする場合は、大量の RAM が必要になります。
- 一部の最適化オプションには、アプリケーションを実行するプロセッサーの種類に関する制限があります。詳細は、オプションの説明を参照してください。

先頭へ戻る

インテル® C++ コンパイラーの使用方法

コマンドラインおよび Linux* からのインテル® C++ コンパイラーの使用方法は、「インテル® Parallel Studio XE 2019: インテル® C++ コンパイラー 19.0 for Linux* 入門」 (*<install-dir>*/documentation_2019/ja/compiler_c/ps2019/get_started_lc.htm) を参照してください。

インテル® C++ コンパイラー for Linux* は、環境モジュール・ソフトウェア・ユーティリティーとともに使用できますが、「モジュールファイル」は含まれていません。詳細は、「インテル® 開発ツールでの環境モジュールの使用」 (英語) を参照してください。

先頭へ戻る

ドキュメント

製品ドキュメントへのリンクは、*<install-dir>*/documentation_2019/ja/compiler_c/ps2019/get_started_lc.htm にあります。すべてのツール・コンポーネントの英語ドキュメントは、インテル® Parallel Studio XE サポートページ (英語) から入手できます。

インストール・イメージからオフライン・コア・ドキュメントを削除

インテル® Parallel Studio XE のインストール・イメージからオフライン・コア・ドキュメントが削除されました。インテル® Parallel Studio XE のコンポーネントのコア・ドキュメントは、インテル® ソフトウェア・ドキュメント・ライブラリー (英語) からオンラインで参照できます。また、インテル® ソフトウェア開発製品レジストレーション・センターから、日本語ドキュメントを含むオフライン・ドキュメントをダウンロードすることもできます: **Product List > Intel® Parallel Studio XE Documentation**。

日本語のサポート

日本語対応のインテル® コンパイラーをインストールした場合、オプションで日本語のサポートが提供されます。エラーメッセージ、仮想開発環境のダイアログ、一部のドキュメントが (英語に加えて) 日本語で提供されます。デフォルトでは、エラーメッセージとダイアログの言語はオペレーティング・システムの言語で表示されます。

日本語のサポートは、すべてのアップデートではなく、一部のアップデートで提供されます。

先頭へ戻る

インテルが提供するデバッグ・ソリューション

- インテルが提供するデバッグ・ソリューションは GNU* GDB ベースです。詳細は、「インテル® Parallel Studio XE 2019 Composer Edition for C++ - デバッグ・ソリューション・リリースノート」(英語) を参照してください。

先頭へ戻る

サンプル

製品のサンプルは、「インテル® ソフトウェア製品のサンプルとチュートリアル」(英語) からダウンロードできます。

先頭へ戻る

テクニカルサポート

インストール時に製品の登録を行わなかった場合は、インテル® ソフトウェア開発製品レジストレーション・センター (http://registrationcenter.intel.com) で登録してください。登録を行うことで、サポートサービス期間

中 (通常は 1 年間)、製品アップデートと新しいバージョンの入手を含む無償テクニカルサポートが提供されます。

テクニカルサポート、製品のアップデート、ユーザーフォーラム、FAQ、ヒント、およびその他のサポート情報は、http://www.intel.com/software/products/support/ (英語) を参照してください。
注: 販売代理店がこの製品のテクニカルサポートを提供している場合、インテルではなく販売代理店にお問い合わせください。

先頭へ戻る

新機能と変更された機能

このバージョンでは、次の機能が新たに追加または大幅に拡張されています。これらの機能に関する詳細は、ドキュメントを参照してください。

#pragma omp simd 向けの精度に影響しない simd オプション

現在、「#pragma omp simd」は浮動小数点値と浮動小数点例外の精度に影響しない設定をオーバーライドします。次のオプションは、SIMD ループでも精度に影響しない値と例外を生成するようにこの動作を変更します。

- qsimd-honor-fp-model[-]: 選択されている浮動小数点モデルに従って SIMD ループをベクトル化します。
- qsimd-serialize-fp-reduction[-]: SIMD ループをベクトル化する際に浮動小数点リダクションをシリアル化します。

OpenMP* SIMD 仕様と FP モデルフラグは、この要件に関して矛盾することがあります。コンパイラーはデフォルトで OpenMP* 仕様に従ってループをベクトル化します。この新しいフラグを使用することで、プログラマーはコンパイラーが FP モデルフラグに従ってループをシリアル化するように設定をオーバーライドできます。
注 1: /Qsimd-honor-fp-model が使用され、OpenMP* SIMD リダクションがループ全体のシリアル化の唯一の原因である場合、/Qsimd-serialize-fp-reduction を使用すると、シリアル化されるリダクション処理を除くループ全体がベクトル化されます。
注 2: このオプションは、ループの自動ベクトル化には影響しません。

-[Q]x / -[Q]ax / -[m]tune / -[m]arch オプションで新しい開発コード名をサポート

サポートされた開発コード名: cascadelake、kabylake、coffeelake、amberlake、whiskeylake。

カスタム・メモリー・アロケーター・ライブラリー

インテル® C++ コンパイラー 19.0 Update 1 では、新しいライブラリー「libqkmalloc」が追加されています。このライブラリーは、メモリー割り当ての C レベルのインターフェイスである「qkmalloc()」を提供します。この割り当ては、標準のルーチンを置き換えることができます。

- void* malloc(size_t size)
- void free(void* ptr)
- void* calloc(size_t nobj, size_t size)
- void* realloc(void* ptr, size_t size)

また、C99 規格に準拠していないメモリー割り当てルーチンであり、「厳密ではない」アライメントを使用する void* _wmalloc(size_t size) を提供します。
このライブラリーを使用するには、次の操作を行います。

- libqkmalloc.so のパスを LD_PRELOAD に設定します。
  ```
  LD_PRELOAD=$INSTALL_DIR/compiler/linux/libqkmalloc.so
  ```
- または、リンカーオプションを定義します。ライブラリーの場所と名前を指定する -L と -l を追加します。
  ```
  -L$QKMALLOC_LIB_PATH -lqkmalloc
  ```

  実行時に LD_LIBRARY_PATH 環境変数にライブラリー・パスを設定します。
  ```
  LD_LIBRARY_PATH="$LD_LIBRARY_PATH:$QKMALLOC_LIB_PATH"
  ```

注: インテル以外のマイクロプロセッサーは、標準のシステム malloc を実行します。

vector プラグマの nodynamic_align 節と vectorlength 節

- 明示的な動的アライメント構文
  ```
  #pragma vector dynamic_align[(pointer)]
  #pragma vector nodynamic_align
  ```

  ポインターを指定しない場合、コンパイラーは通常どおりに振る舞います (アライメントする必要があるポインターを自動的に判断するか、ピールループを生成しません)。ポインターを指定すると、コンパイラーはそのポインターのピールループを生成します。nodynamic_align 節を指定すると、コンパイラーはピールループを生成しません。

- ```
  #pragma vector vectorlength(vl1,vl2, .., vln)
  #pragma vector vectorlength(vl1,vl2, .., vln)
  ```

  ベクトライザーは、コストモデルに応じてリストから最適なベクトル長を選択します。リストに最適なベクトル長がない場合、ループはスカラーのままとなります。このプラグマはベクトル化を強制しないため、すべてのループに対して安全に使用できます。

OpenMP* TR7 Version 5.0 Draft の機能

OpenMP* Technical Report 6: Version 5.0 Preview 2 (英語) 仕様の言語機能がサポートされました。

- 明示的な包括スキャン構文*
  ```
  #pragma omp simd reduction[parallel](inscan, operator:list)
  #pragma omp scan inclusive(item-list)
  ```
- 明示的な排他スキャン構文*
  ```
  #pragma omp simd reduction[parallel](inscan, operator:list)
  #pragma omp scan exclusive(item-list)
  ```
  プリフィクス sum は、ベクトル実行中に正しく計算されます。
  ```

- OpenMP* 並列プラグマの UDI
  ```
  #pragma omp declare induction ( induction-id : induction-type :step-type : inductor ) [collector( collector )]
  ```

詳細は、コンパイラー・ドキュメントまたは上記の OpenMP* 仕様へのリンクを参照してください。

C++17 の機能をサポート

インテル® C++ コンパイラー 19.0 は、*/Qstd=c++17* (Windows*) または *-std=c++17* (Linux*/macOS*) オプションで次の機能をサポートします。

- 畳み込み式 (N4295 (英語))
- インライン変数 (P0386R2 (英語))
- 列挙型クラスの構文規則 (P0138R2 (英語))
- 廃止予定の動的例外仕様の削除 (P0003R5 (英語))
- 例外仕様を型システムの一部にする (P0012R1 (英語))
- constexpr ラムダ式 (P0170R1 (英語))
- *this のラムダ・キャプチャー (P0018R3 (英語))
- if constexpr 文 (P0292R2 (英語))
- 構造化バインド(P0217R3 (英語))
- if 文と switch 文の変数と条件を分離 (P0305R1 (英語))
- 以前の主要バージョンのコンパイラーとの比較を含む、サポートしている機能の最新リストは、「インテル® C++ コンパイラーでサポートされる C++17 の機能」を参照してください。

C++14 の機能をサポート

インテル® C++ コンパイラー 19.0 は、*/Qstd=c++14* (Windows*) または *-std=c++14* (Linux*/macOS*) オプションで次の機能をサポートします。

- 以前の主要バージョンのコンパイラーとの比較を含む、サポートしている機能の最新リストは、「インテル® C++ コンパイラーでサポートされる C++14 の機能」を参照してください。

C++11 の機能をサポート

インテル® C++ コンパイラー 19.0 は、*/Qstd=c++11* (Windows*) または *-std=c++11* (Linux*/macOS*) オプションで次の機能をサポートします。

- 以前の主要バージョンのコンパイラーとの比較を含む、サポートしている機能の最新リストは、「インテル® C++ コンパイラーでサポートされる C++11 の機能」を参照してください。

C11 の機能をサポート

インテル® C++ コンパイラーは、*/Qstd=c11* (Windows*) または *-std=c11* (Linux*/macOS*) コンパイラー・オプションで以下の C11 の機能をサポートします。

- 以前の主要バージョンのコンパイラーとの比較を含む、サポートしている機能の最新リストは、「インテル® C++ コンパイラーにおける C11 サポート」を参照してください。

新規および変更されたコンパイラー・オプション

コンパイラー・オプションの詳細は、『インテル® C++ コンパイラー 19.0 デベロッパー・ガイドおよびリファレンス』の「コンパイラー・オプション」セクションを参照してください。

- -qopenmp-simd をデフォルトで設定
- 新しい -xcannonlake オプション
- 新しい -mtune=cannonlake オプション
- -rcd オプションは、切り捨ての代わりに最近値への丸めを使用することで、「高速な」浮動小数点から整数への変換を可能にします。これは廃止予定のオプションです。

廃止予定のコンパイラー・オプションのリストは、『インテル® C++ コンパイラー 19.0 デベロッパー・ガイドおよびリファレンス』の「コンパイラー・オプション」セクションを参照してください。

先頭へ戻る

C++ STL の並列およびベクトル実行向け Parallel STL

インテル® C++ コンパイラーとともに、ポリシーの実行をサポートする C++ 標準ライブラリーのアルゴリズムの実装である Parallel STL がインストールされます。

機能/API の変更点

- さらに多くのアルゴリズムで並列およびベクトル実行ポリシーをサポート: find_first_of、is_heap、is_heap_until、replace、replace_if。
- さらに多くのアルゴリズムでベクトル実行ポリシーをサポート: remove、remove_if。
- さらに多くのアルゴリズムで並列実行ポリシーをサポート: partial_sort。

詳細は、https://software.intel.com/en-us/get-started-with-pstl (英語) を参照してください。

終了予定のサポート

インテル® Cilk™ Plus は 18.0 以降では非推奨

インテル® Cilk™ Plus は、インテル® C++ コンパイラー 18.0 以降では非推奨の古い機能です。詳細は、「インテル® Cilk™ Plus アプリケーションを OpenMP* もしくはインテル® TBB へ移行する」を参照してください。

終了したサポート

インテル® グラフィックス・テクノロジーへのオフロードサポートは終了しました。

先頭へ戻る

既知の制限事項

Parallel STL

unseq および par_unseq ポリシーは、'#pragma omp simd' または '#pragma simd' をサポートするコンパイラーでのみ有効です。並列およびベクトル実行は、ランダム・アクセス・イテレーターが提供される場合にのみアルゴリズムのサブセットでサポートされ、残りはシリアル実行のままとなります。コンパイラーによっては、zip_iterator は unseq および par_unseq ポリシーで動作しないことがあります。

ポインターチェッカーにダイナミック・ランタイム・ライブラリーが必要

-check-pointers オプションを使用する場合は、ランタイム・ライブラリー `libchkp.so` をリンクする必要があります。*-static* または *-static-intel* のようなオプションを *-check-pointers* とともに使用すると、設定に関係なくこのダイナミック・ライブラリーがリンクされることに注意してください。詳細は、「ICC のポインターチェッカー」(英語) を参照してください。

先頭へ戻る

著作権と商標について

最適化に関する注意事項
インテル® コンパイラーでは、インテル® マイクロプロセッサーに限定されない最適化に関して、他社製マイクロプロセッサー用に同等の最適化を行えないことがあります。これには、インテル® ストリーミング SIMD 拡張命令 2、インテル® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサーに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサー依存の最適化は、インテル® マイクロプロセッサーでの使用を前提としています。インテル® マイクロアーキテクチャーに限定されない最適化のなかにも、インテル® マイクロプロセッサー用のものがあります。この注意事項で言及した命令セットの詳細については、該当する製品のユーザー・リファレンス・ガイドを参照してください。 注意事項の改訂 #20110804

切責任を負いません。この情報は予告なく変更されることがあります。この情報だけに基づいて設計を最終的なものとしないでください。

本資料で説明されている製品には、エラッタと呼ばれる設計上の不具合が含まれている可能性があり、公表されている仕様とは異なる動作をする場合があります。現在確認済みのエラッタについては、インテルまでお問い合わせください。

最新の仕様をご希望の場合や製品をご注文の場合は、お近くのインテルの営業所または販売代理店にお問い合わせください。

本資料で紹介されている資料番号付きのドキュメントや、インテルのその他の資料を入手するには、1-800-548-4725 (アメリカ合衆国) までご連絡いただくか、http://www.intel.com/design/literature.htm (英語) を参照してください。

インテル・プロセッサー・ナンバーはパフォーマンスの指標ではありません。プロセッサー・ナンバーは同一プロセッサー・ファミリー内の製品の機能を区別します。異なるプロセッサー・ファミリー間の機能の区別には用いません。詳細については、http://www.intel.co.jp/jp/products/processor_number/ を参照してください。

インテル® C++ コンパイラーは、インテルのソフトウェア使用許諾契約書 (EULA) の下で提供されます。

詳細は、製品に含まれるライセンスを確認してください。

Intel、インテル、Intel ロゴ、Cilk は、アメリカ合衆国および / またはその他の国における Intel Corporation またはその子会社の商標です。

* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

先頭へ戻る