インテル® C++ コンパイラー 18.0 for Linux* リリースノート (インテル® Parallel Studio XE 2018)

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 18.0 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

Update 3 (インテル® C++ コンパイラー 18.0.3)

- 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 -mconditional-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的実行のサイドチャネル問題を軽減する」(英語) を参照してください。
- __INTEL_LIBIRC_DEBUG 環境変数

Update 2 (インテル® C++ コンパイラー 18.0.2)

- 特定の CPU の指定で新しい開発コード名をサポート
- 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的実行のサイドチャネル問題を軽減する」(英語) を参照してください。
- 報告された問題を修正

Update 1 (インテル® C++ コンパイラー 18.0.1)

- -static-intel が指定された場合も含め、インテルの OpenMP* ランタイム・サポート・ライブラリーを動的にリンク。OpenMP* ライブラリーを静的にリンクするには、-qopenmp-link=static を指定します。18.0.0 のデフォルトの動作は正しくなく、修正されました。
- 日本語版を含む最初のアップデート
- 報告された問題を修正

インテル® C++ コンパイラー 17.0 以降 (インテル® C++ コンパイラー 18.0 での変更)

- 新しいオプション -qopt-zmm-usage
- Control-Flow Enforcement Technology (CET) サポート
- SVML を強制する新しいオプション -fimf-use-svml
- SVML 呼び出しをコンパイル時に割り当て
- インテル® Xeon Phi™ x100 製品ファミリー (開発コード名 Knights Corner) のサポート終了
- すべての -o* オプションを -qo* オプションに変更
- C++ STL の並列およびベクトル実行向け Parallel STL
- ハードウェア・ベースの PGO サポート
- simd 構文の ordered ブロック向け monotonic および overlap キーワード
- 抽出組込み関数 (_mm256_extract_epi8) の戻り型を変更
- OpenMP* TR4 Version 5.0 Preview 1 の機能
- OpenMP* 4.0 以降の新機能をサポート
- 新しい C++17 機能をサポート
- C11 機能の atomic キーワードをサポート
- 新規および変更されたコンパイラー・オプション
- 32 ビット icc ラッパーは 18.0 では非推奨
- インテル® Cilk™ Plus は 18.0 では非推奨
- インストール・イメージからオフライン・ドキュメントを削除

先頭へ戻る

動作環境

- インテル® ストリーミング SIMD 拡張命令 2 (インテル® SSE2) 対応のインテル® 64 アーキテクチャー・プロセッサーをベースとするコンピューター (第 2 世代以降のインテル® Core™ i3/i5/i7 プロセッサー、インテル® Xeon® プロセッサー E3/E5 ファミリー、または互換性のあるインテル以外のプロセッサー)
  - 64 ビット・アプリケーションおよびインテル® メニー・インテグレーテッド・コア (インテル® MIC) 向けのアプリケーションの開発は、64 ビット・バージョンの OS でのみサポートしています。32 ビット・アプリケーションの開発も、64 ビット・バージョンの OS でのみサポートしています。コンパイラーは、32 ビットの OS にインストールできません。
  - 64 ビット・バージョンの OS で 32 ビット向けのアプリケーションを開発する場合は、Linux* ディストリビューションからオプションのライブラリー・コンポーネント

(ia32-libs、lib32gcc1、lib32stdc++6、libc6-dev-i386、gcc-multilib、g++-multilib) をインストールする必要があります。

- インテル® MIC アーキテクチャー向けの開発/テスト
  - インテル® メニーコア・プラットフォーム・ソフトウェア・スタック (インテル® MPSS)
- インテル® グラフィックス・テクノロジーへのオフロードまたはネイティブサポートの開発/テスト
  - オフロードは 64 ビット・アプリケーションでのみサポートされます。
  - インテル® グラフィックス・テクノロジー対応 64 ビット・グラフィックス・ドライバー (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) から入手できます)。インテル® Parallel Studio XE を登録すると Linux* 用インテル® HD グラフィックス・ドライバーのダウンロード情報にアクセスできるようになります。この情報にアクセスできない場合は、テクニカルサポート (英語) までお問い合わせください。次のドライバーバージョン、対応するオペレーティング・システム、およびプロセッサーをサポートしています。
  - Linux* 用インテル® HD グラフィックス・ドライバー 16.4.4 (第 4 世代および第 5 世代インテル® Core™ プロセッサー用)
  - CentOS* 7.1 (64 ビット・アーキテクチャー)
  - Red Hat* Enterprise Linux* 7.1 (64 ビット・アーキテクチャー)
  - 次のプロセッサーをサポートしています。
    - 第 5 世代インテル® Core™ プロセッサー (インテル® Iris® グラフィックス、インテル® HD グラフィックス 5500/6000/6100)
    - 第 4 世代インテル® Core™ プロセッサー (インテル® Iris® Pro グラフィックス、インテル® Iris® グラフィックス、またはインテル® HD グラフィックス 4200+ 番台) (チップセットの互換性は通常、インテル® Core™ プロセッサーでは問題になりません)
    - インテル® Xeon® プロセッサー E3 v3 ファミリー (インテル® HD グラフィックス P4700)
    - インテル® Xeon® プロセッサー E3 v4 ファミリー (インテル® Iris® Pro グラフィックス P6300)
    - **注意事項:**
      - チップセットでプロセッサー・グラフィックスを有効にする必要があります。データシートを必ず確認してください。
      - インテル® Xeon® プロセッサーにはインテル® C226 チップセットが必要です。
      - 第 4 世代より前のインテル® Core™ プロセッサーはサポートしていません。
      - インテル® Celeron® プロセッサー、インテル® Pentium® プロセッサーおよび Intel Atom® プロセッサーはサポートしていません。
- 機能を最大限に活用できるよう、マルチコアまたはマルチプロセッサー・システムの使用を推奨します。
- 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* 7.0、8.0、9.0
    - Fedora* 24、25、26
    - Red Hat* Enterprise Linux* 6、7
    - SUSE* Linux* Enterprise Server 11、12
    - Ubuntu* 14.04 LTS、15.10、16.04 LTS、16.10、17.04 LTS
    - 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 - 6.3 をサポート
    - binutils バージョン 2.20-2.26 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 4.6 および次の両方
    - Eclipse* C/C++ Development Tools (CDT) 8.8 または 9.0
    - Java* ランタイム環境 (JRE) 8.0 (1.8) 以降
- Eclipse* Platform 4.5 および次の両方
    - Eclipse* C/C++ Development Tools (CDT) 8.7
    - Java* ランタイム環境 (JRE) 7.0 (1.7) 以降
- Eclipse* Platform 4.4 および次の両方
    - Eclipse* C/C++ Development Tools (CDT) 8.4-8.6
    - Java* ランタイム環境 (JRE) 7.0 (1.7) 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。
- 上記のリストにはすべてのプロセッサー・モデル名は含まれていません。リストされているプロセッサーと同じ命令セットを正しくサポートしているプロセッサー・モデルでも動作します。特定のプロセッサー・モデルについては、テクニカルサポートにお問い合わせください。
- 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。
- インテルは、ユーザーの課題に最適なソリューションを提供するため、インテル製品の市場を継続的に評価しています。この評価プロセスの一環として、インテルはインテル® Xeon Phi™ コプロセッサー 7200 (開発コード名 Knights Landing) を市場に投入しないことを決定しました。

- インテル® Xeon Phi™ 7200 プロセッサーの迅速な採用状況を考慮し、インテルは一般市場に Knights Landing (開発コード名) コプロセッサーを展開しないことを決めました。
- インテル® Xeon Phi™ プロセッサーは、引き続きインテルのソリューション・ポートフォリオの主要要素として、ユーザーに最も魅力的で競争力のあるソリューションを提供します。

インテル® メニーコア・プラットフォーム・ソフトウェア・スタック (インテル® MPSS)

インテル® メニーコア・プラットフォーム・ソフトウェア・スタック (インテル® MPSS) は、インテル® C++ コンパイラーのインストール前またはインストール後にインストールできます。
最新バージョンのインテル® MPSS を使用することを推奨します。インテル® Parallel Studio XE for Linux* を登録すると、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) から入手できます。
ユーザー空間およびカーネルドライバーのインストールに必要な手順については、インテル® MPSS のドキュメント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 2018: インテル® C++ コンパイラー 18.0 for Linux* 入門」(*<install-dir>*/documentation_2018/ja/compiler_c/ps2018/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_2018/ja/compiler_c/ps2018/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。日本語ドキュメントは、ドキュメントの ja サブディレクトリーに含まれています。

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 2018 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。

__INTEL_LIBIRC_DEBUG 環境変数

OS/HW が正しく動作せず、CPUID の libirc のキャッシュサイズが 0 の場合、パフォーマンスが低下する可能性があります。その場合、環境変数 __INTEL_LIBIRC_DEBUG=1 を定義して、原因を特定し、OS/HW 問題に関する情報を取得できます。

例: export __INTEL_LIBIRC_DEBUG=1

特定の CPU の指定で新しい開発コード名をサポート

次の新しい cpuid (開発コード名) をサポート

- haswell
- broadwell
- skylake
- skylake_avx512
- knl
- knm

次の関数属性で利用可能 (関数を特定 の cpu 開発コード名向けに手動で指定):

```
C/C++ __attribute__((cpu_dispatch()))
C/C++ __attribute__((cpu_specific()))
C/C++ __attribute__((vector(processor())))
```

simd 構文の ordered ブロック向け monotonic および overlap キーワード

既存の `#pragma omp ordered simd` 用の新しいキーワード

- `#pragma omp ordered simd monotonic()`
- `#pragma omp ordered simd overlap(expr)`
- `#pragma omp simd reduction(=: list)`

詳細は、『インテル® C++ コンパイラー 18.0 デベロッパー・ガイドおよびリファレンス』を参照してください。

ハードウェア・ベースの PGO サポート

プロファイルに基づく最適化 (PGO) のハードウェア・ベースのイベント・サンプリングは、インテル® コンパイラーとインテル® VTune™ Amplifier を利用して PGO の多くの利点を得ることができる、新しい低オーバーヘッドのモデルです。インテル® VTune™ Amplifier がサポートされているシステムでデータを収集することができます。

詳細は、『インテル® C++ コンパイラー 18.0 デベロッパー・ガイドおよびリファレンス』を参照してください。

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。

Parallel STL の詳細は、入門 (英語) およびリリースノート (英語) を参照してください。

抽出組込み関数 (_mm256_extract_epi8) の戻り型を変更

256 ビット・ベクトル組込み関数 `_m256_extract_epi8/epi16(__m256i a, const int index)` が __int8/__int16 値の代わりに整数値を返すように定義されました。

SVML を強制する新しいオプション -fimf-use-svml

新しいオプションは、スカラー数学演算に LIBM が使用されている場合、SVML の使用を強制します。これにより、ベクトルコードとスカラーコードの計算結果がビット単位で同じになることが保証されます。この機能を利用して、コンパイラーは -fp:precise FP モデルで数学関数をベクトル化し、ベクトル化されたコードはスカラーコードと同じ結果を生成します。

新しいオプション -qopt-zmm-usage

新しいオプション -qopt-zmm-usage:low|high を使用して、コンパイラーにより生成される zmm コードをチューニングできます。引数値 low は、インテル® Xeon® Platinum プロセッサー (開発コード名 Skylake) 上のエンタープライズ・アプリケーションなどで、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) ISA からインテル® 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) ISA へのスムーズな移行を可能にします。ZMM 命令向けのチューニングには、#pragma omp simd simdlen() などの明示的なベクトル構文を使用することを推奨します。引数値 high は、幅広いベクトル操作を利用して命令ごとの計算量を高める、ベクトル演算が主体の HPC コードなどのアプリケーションに適しています。デフォルト値は、Skylake Server (開発コード名) マイクロアーキテクチャーをターゲットとする場合は low で、インテル® Core™ マイクロアーキテクチャー/インテル® メニー・インテグレーテッド・コア (インテル® MIC) アーキテクチャーのインテル® AVX-512 をターゲットとする場合は high です。

Control-Flow Enforcement Technology (CET) サポート

Control-flow Enforcement Technology (CET) は、ROP (Return-Oriented Programming) や COP/JOP (Call/Jump-Oriented Programming) などの脆弱性を悪用する特定の攻撃からプログラムを防御します。詳細は、プレビュー・ドキュメント (英語) を参照してください。

新しいコンパイラー・オプション -cf-potection[:keyword] により、コンパイラーの CET サポートを有効にできます。

SVML 呼び出しをコンパイル時に割り当て

SVML 関数に対するコンパイラーのデフォルトの動作が変更され、CPU 固有の SVML エントリーの呼び出しが実行されるようになりました。新しいオプション -fimf-force-dynamic-target を指定すると、以前の動作に戻して動的に SVML をディスパッチできます。

OpenMP* TR4 Version 5.0 Preview 1 の機能

OpenMP* Technical Report 4 : Version 5.0 Preview 1 (英語) 仕様のタスク・リダクション言語機能がサポートされました。

- TASKGROUP に TASK_REDUCTION 節が追加されました。
- TASK に IN_REDUCTION 節が追加されました。
- TASKLOOP に REDUCTION 節と IN_REDUCTION 節が追加されました。

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。

OpenMP* 4.0 以降の新機能をサポート

- taskloop 構文 `#pragma omp taskloop[clause[[,]clause]..]` をサポート
- *#pragma omp for linear (list [: linear-step])* をサポート
  - *list* は *list* または *modifier(list)* のいずれか
- *linear* 節で *ref*、*val*、*uval* 修飾子をサポート
  - 例: *linear(ref(p)), linear(val(i):1), linear(uval(j):1)*
- *#pragma omp simd simdlen(n)* をサポート
- *#pragma omp ordered [simd]* をサポート
- 配列全体のリダクション: *int x[n]; #pragma omp simd reduction(+:x)*
- *processor* 節の拡張を *#pragma omp declare simd* に追加 (OpenMP* 4.5 の正式な機能ではありません)
- `#pragma omp for schedule` 節の `SIMD` 修飾子と `NONMONOTONIC` 修飾子をサポート
  - インテル® C++ コンパイラー 18.0 には、ループの反復をチームのスレッド間でどのように分割するかについて、ユーザー制御を強化する `SIMD` 修飾子と `NONMONOTONIC` 修飾子の拡張が含まれています。詳細は、『インテル® C++ コンパイラー・デベロッパー・ガイドおよびリファレンス』を参照してください。
- reduction 節で部分配列をリスト項目としてサポート
  - `reduction(reduction-identifier:list)`: リスト項目が部分配列の場合、reduction 節がセクションの個々の要素に適用されると見なして処理されます。プライベートの部分配列の要素は、連続して割り当てられます。

C++17 の機能をサポート

インテル® C++ コンパイラー 18.0 は、*/Qstd=c++17* (Windows*) または *-std=c++17* (Linux*/macOS*) コンパイラー・オプションで以下の C++17 の機能をサポートします。

- static_assert のメッセージ省略を許可 (N3928)
- 範囲 for ループの制限緩和 (N3994)

- 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。

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。

- _Atomic および __attribute((atomic)) キーワードを含むすべての C11 の機能
- 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 18.0 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。

- **すべての -o* オプションを -qo* オプションに変更**

  以前のバージョンで非推奨とされていたすべての -o* オプションが、このバージョンでは -qo* オプションに置き換えられました。ただし、Linux* と macOS* で出力ファイルを変更する -o オプションは変更されません。

  Windows* では、この変更は /Qoffload-option コンパイラー・オプションでターゲット・コンパイラーに渡されるコンパイラー・オプションに影響します。

  古い -o* オプションを使用すると、次のようなコンパイラー診断が出力されます。

  ```
  $ icc -openmp example.c
  icc: コマンドライン・エラー: オプション '-openmp' はサポートされていません。オプション '-qopenmp' を使用してください。
  ```

  影響のあるオプション:

  -[no-]openmp
  -openmp-lib=<arg>
  -openmp-link=<arg>
  -[no-]openmp-offload
  -[no-]openmp-simd
  -openmp-stubs
  -openmp-threadprivate=<arg>
  -openmp-report[=<level>]
  -openmp-task=<arg>
  -opt-args-in-regs=<arg>
  -[no-]opt-assume-safe-padding
  -opt-block-factor=<arg>

-[no-]opt-calloc
-[no-]opt-class-analysis
-[no-]opt-dynamic-align
-[no-]opt-gather-scatter-unroll
-[no-]opt-jump-tables=<arg>
-opt-malloc-options=<arg>
-[no-]opt-matmul
-[no-]opt-mem-layout-trans=<arg>
-[no-]opt-multi-version-aggressive
-[no-]opt-prefetch[=<val>]
-opt-prefetch-distance=<arg>
-opt-ra-region-strategy[=<arg>]
-[no-]opt-report-embed
-opt-report-file=<arg>
-opt-report-filter=<arg>
-opt-report-format=<arg>
-opt-report-phase=<arg>
-opt-report-routine=<arg>
-opt-report-help
-opt-report[=<arg>]
-opt-report-per-object
-opt-streaming-cache-evict=<arg>
-opt-streaming-stores=<arg>
-[no-]opt-subscript-in-range
-opt-threads-per-core=<arg>

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 18.0 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

終了予定のサポート

インテル® Cilk™ Plus は 18.0 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 では非推奨の古い機能です。プロセッサー・グラフィックスへのオフロードには、OpenMP* ベースの構文を使用することを推奨します。詳細は、「インテル® Cilk™ Plus の代わりに OpenMP* またはインテル® TBB を使用するためのアプリケーションの移行」を参照してください。

32 ビット icc ラッパーは 18.0 では非推奨

icc: リマーク #10421: IA-32 ターゲット・ラッパー・バイナリー 'icc' は古いオプションです。コンパイラー・スタートアップ・スクリプトまたは適切なインテル(R) 64 コンパイラー・バイナリーで '-m32' オプションを使用して目的のアーキテクチャーをターゲットにしてください。

終了したサポート

インテル® Xeon Phi™ x100 製品ファミリー (開発コード名 Knights Corner) のサポート終了

- Visual Studio* の [Code Generation [Intel C++] (コード生成 [インテル(R) C++])] > [Offload Target Architecture (オフロード・ターゲット・アーキテクチャー)] から (/Qoffload-Arch): "mic" が削除されました。
- [Platform Toolset (プラットフォーム・ツールセット)] が "Intel C++ Compiler 17.0" に設定されているプロジェクトで "mic" 値が指定されている場合、"Intel C++ Compiler 18.0" プラットフォーム・ツールセットへアップグレード後に "既定値" に更新されます。

IA-32 ホストへのインストールのサポートを終了

IA-32 ホストへのインストールのサポートを終了しました。32 ビット・ターゲット用コード生成のサポートは 64 ビット・ホストでサポートされます (*-m32* コンパイラー・オプションを使用)。

Red Hat* Enterprise Linux* 5 のサポートを終了

このオペレーティング・システム・バージョンのサポートを終了しました。新しいバージョンのオペレーティング・システムに移行してください。

第 3 世代インテル® Core™ プロセッサーの GFX オフロードのサポートを終了

第 3 世代インテル® Core™ プロセッサーのプロセッサー・グラフィックスへの HW オフロードのサポートは、インテル® C++ コンパイラー 17.0 で終了しました。

Linux* 用インテル® HD グラフィックス・ドライバー 16.3.2 (第 3 世代および第 4 世代インテル® Core™ プロセッサー用) のサポートを終了

16.3.2 ドライバーのサポートを終了しました。第 4 世代インテル® Core™ プロセッサー・ベースのシステムは、16.4.2 ドライバーおよびこのドライバーをサポートしているオペレーティング・システム (CentOS* 7.1 または RHEL 7.1) に移行することを推奨します。

32 ビット・ホストへのインストールのサポートを終了

32 ビット・ホストへのインストールのサポートは、このリリースで終了しました。32 ビット・ターゲット用コードの生成は 64 ビット・ホストでサポートされます (*-m32* コンパイラー・オプションを使用)。

***_GFX_enqueue* を削除**

_GFX_enqueue は削除されました。*_GFX_offload* に変更してください。

先頭へ戻る

既知の制限事項

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、http://intel.ly/1jV0eWD (英語) を参照してください。

インテル® メニー・インテグレーテッド・コア (インテル® MIC) アーキテクチャーの既知の問題

- ランタイム型情報 (RTTI) は未サポート

  仮想共有メモリー・プログラミングでは、ランタイム型情報 (RTTI) はサポートされていません。特に、*dynamic_cast<>* と *typeid()* の使用はサポートされていません。

インテル® グラフィックス・テクノロジーへのオフロードの既知の問題

- オフロードコードのホストバージョンが並列化されない

  コンパイラーは、#pragma offload 以下に並列ループのターゲットバージョンとホストバージョンの両方を生成します。ホストバージョンは、オフロードが実行できない場合 (通常は、ターゲットシステムにインテル® グラフィックス・テクノロジーが有効なユニットがない場合) に実行されます。並列ループは、オフロードの並列セマンティクスを含む cilk_for の並列構文または配列表記文を使用して指定する必要があります。ターゲットバージョンはターゲット実行で並列化されますが、現在、ホスト側のバックアップ・バージョンが並列化されない制限があります。そのため、cilk_for を使用する場合、オフロード実行が行われないと、バックアップ・コードの実行パフォーマンスに大きく影響する可能性があることに注意してください。配列表記文は現在ホスト側で並列コードを生成しないため、パフォーマンスに影響はありません。これは既知の問題で、将来のリリースで修正される予定です。

- 非 root 権限で複数のプロセスを実行するとオフロードに失敗することがある

  (非 root 権限で) 複数のプロセスをオフロードしようとすると失敗することがあります。/dev/dri/card0 を開く最初のプロセスのみ DRM 認証をパスすることができ、master 権限があります。DRM 認証をパスするには、"root" または "master" 権限が必要です。このため、root 権限で実行するとすべてのプロセスがパスしますが、非 root 権限では 1 つのプロセスしかパスしません。これは Linux* の既知の制限です。

  回避策は、次のいずれかです。

  - 各プロセスをシリアルに実行します。
  - root として実行します。

- インテル® グラフィックス・テクノロジーへのオフロードの既知の制限事項
  - オフロードコードでは、次の機能を使用できません。

        - 例外処理
        - RTTI
        - longjmp/setjmp
        - VLA
        - 変数引数リスト
        - 仮想関数、関数ポインター、その他の間接呼び出しまたはジャンプ
        - 共有仮想メモリー
        - 配列や構造体のようなポインターを含むデータ構造
        - ポインターまたは参照型のグローバル変数
        - OpenMP*
        - cilk_spawn または cilk_sync
        - インテル® Cilk™ Plus のレデューサー
        - ANSI C ランタイム・ライブラリー呼び出し (SVML、math.h、mathimf.h 呼び出し、およびその他いくつかの例外あり)
    - 64 ビット浮動小数点演算および整数演算は非効率

インテル® Cilk™ Plus の既知の問題

- ランタイムのスタティック・リンクはサポートされていません。

  インテル® Cilk™ Plus ライブラリーのスタティック・バージョンは、意図的に提供されていません。スタティック・ライブラリーをリンクする *-static-intel* を使用すると、警告が表示され、インテル® Cilk™ Plus ライブラリーのダイナミック・バージョン (*libcilkrts.so*) がリンクされます。

  ```
  $ icc -static-intel sample.c

  icc: 警告 #10237: -lcilkrts はダイナミックにリンクされました。スタティック・ライブラリーは利用できません。
  ```

  別の方法として、スタティック・ランタイムを使用してインテル® Cilk™ Plus のオープンソース・バージョンをビルドできます。インテル® Cilk™ Plus の実装についての詳細は、https://www.isus.jp/article/intel-cilk-plus/ を参照してください。インテル® Cilk™ Plus ライブラリーのダイナミック・バージョンで問題が発生した場合は、テクニカルサポートまでご連絡ください。

IA-32 アプリケーション用インテル® C++ コンパイラーを Red Hat* Enterprise Linux* 6 で使用すると SPEC* CPUv6 ランタイム・バス・エラーが発生する

- SPEC* CPUv6 ベンチマーク (現在開発中、英語) を IA-32 アプリケーション用インテル® C++ コンパイラーでコンパイルした後、*ulimit* コマンド (例えば、*ulimit -s 2067152 -v 15000000*) を使用してスタック/仮想メモリーを制限して実行すると、バスエラーが発生します。現在、この問題はこのベンチマークでのみ発生していますが、ほかのアプリケーションでも発生する可能性があります。このエラーを回避するには、これらのパラメーターを *unlimited* に設定してください。

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

本資料に掲載されている情報は、インテル製品の概要説明を目的としたものです。本資料は、明示されているか否かにかかわらず、また禁反言によるとよらずにかかわらず、いかなる知的財産権のライセンスも許諾するものではありません。製品に付属の売買契約書『Intel's Terms and Conditions of Sale』に規定されている場合を除き、インテルはいかなる責任を負うものではなく、またインテル製品の販売や使用に関する明示または黙示の保証 (特定目的への適合性、商品適格性、あらゆる特許権、著作権、その他知的財産権の非侵害性への保証を含む) に関してもいかなる責任も負いません。インテルによる書面での合意がない限り、インテル製品は、インテル製品の欠陥や故障によって人身事故が発生するような用途向けに使用することを前提としたものではありません。

インテル製品は、予告なく仕様や説明が変更されることがあります。機能または命令の一覧で「留保」または「未定義」と記されているものがありますが、その「機能が存在しない」あるいは「性質が留保付である」という状態を設計の前提にしないでください。これらの項目は、インテルが将来のために留保しているものです。インテルが将来これらの項目を定義したことにより、衝突が生じたり互換性が失われたりしても、インテルは一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、Intel Atom、Celeron、Intel Core、Iris、Pentium、Xeon、Intel Xeon Phi、Cilk、VTune 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 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。

© 2018 Intel Corporation. 無断での引用、転載を禁じます。

先頭へ戻る

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。