

インテル® oneAPI DPC++/C++ コンパイラー 2021.1 リリースノート

2020年12月4日

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。

リリースの入手方法

[このサイト](#) (英語) から手順に従ってツールキットをダウンロードし、インストール手順に従ってツールキットをインストールします。

インテル® oneAPI DPC++/C++ コンパイラー 2021.1 の主な機能

DPC++

- [DPC++ 1.0仕様](#) (英語) に準拠
- Ahead-Of-Time (AOT) コンパイルのサポート
- 明示的な SIMD プログラミングの試験的サポート
- SYCL* 標準の進化に合わせて `CL_SYCL_LANGUAGE_VERSION` を `SYCL_LANGUAGE_VERSION` に置換
- Visual Studio* 2017/2019 (Windows*) および Eclipse* (Linux*) との統合
- `-fsycl-device-lib=<value>` (引数は `libc`、`libm-fp32`、`libm-fp64`、`all`) オプションを使用してカーネルコードで `std::*` 数学関数を使用するアプリケーションをコンパイル
- 詳細な情報および利用可能な組み込み関数は、[インタラクティブ組み込み関数ガイド](#) (英語) を参照
- Linux* パッケージ・マネージャー (YUM、APT、Zypper) を使用した oneAPI ベース・ツールキット用インテル® FPGA アドオンのインストールをサポート
- 複数の FPGA プラットフォームのターゲットをサポート

OpenMP* オフロード

- OpenMP* 4.5 および OpenMP* 5.1 サブセットをサポート
- 複数の GPU で OpenMP* オフロードをサポート
- OpenMP* オフロードの `opt-report`
- OpenMP* と DPC++ の相互運用性

- インテルの USM 割り当て API 拡張をサポート
- GPU 実行にインテル® MKL を呼び出すインテルの拡張をサポート
- OpenMP* オフロード領域でインライン v-ISA をサポート

既知の問題と制限事項

- DPC++/C++ コンパイラーの `credist.txt` ファイルは Gold リリースではオンラインでのみ利用可能です。将来のリリースではコンパイラー・パッケージに含まれる予定です。
- サブグループ・アルゴリズムを条件文 (例えば、"if") を含むループで使用すると、CPU 上での結果が正しくないことがあります。
- OpenMP* ターゲットコードで `scalbn()` を使用すると、ランタイムエラーが発生します。この問題を回避するには、`scalbn()` を `ldexp()` に置換します。この問題は、将来のリリースで修正される予定です。
- DPC++ コンパイラーは、ターゲット・オフロード・コードを含むライブラリーで `-l` オプションを使用したライブラリー・アーカイブのリンクをサポートしていません。この問題の詳細および回避策は、「[既知の問題: スタティック・ライブラリーとターゲットオフロード](#)」(英語) を参照してください。
- リンク時の最適化 (LTO) を使用しようとする、一部の Linux* OS でリンクエラーが発生することがあります。正常にリンクするには、「[インテル® oneAPI DPC++/C++ コンパイラーのシステム要件](#)」(英語) にリストされている、OS で推奨するバージョンの `binutils` を使用してください。
- OpenCL* C 組込み関数と同じ名前およびシグネチャー (戻り値の型は関係なく、引数が完全に一致している) のユーザー定義関数を使用すると、未定義の動作を引き起こすことがあります。この問題の詳細は、「[既知の問題: OpenCL* 組込み関数と同じシグネチャーのユーザー定義関数](#)」(英語) を参照してください。
- DPC++ ランタイム・ライブラリーは、「[セマンティック・バージョンング](#)」の規則 (メジャー・マイナー・パッチ) に従っています。メジャーバージョンは後方互換性のない変更を示します (バージョン X はバージョン X-1 と後方互換性がありません)。マイナーバージョンは後方互換性がある変更を示します。回避策は、アプリケーションをリビルドすることです。
- コンパイラーにより生成されるオブジェクト・ファイルの形式は、バージョン間で変わることがあります。回避策は、アプリケーションをリビルドすることです。
- `cl::sycl::program` API を使用して別の翻訳単位で定義されたカーネルを参照すると、未定義の動作を引き起こします。
- 画像アクセサーに読み取りサンプラーを使用すると、レベルゼロプラグイン/バックエンドで問題が発生することがあります。
- 内部定義の出力は Windows* ではサポートしていません。
- `MUL/AND/OR/XOR` のグループ・アルゴリズムは、SPIR-V* の制限によりグループスコープで有効にできません。また、SPIR-V* のバージョンは 1.1 から 1.3 に自動的に変更されないため、まだサブグループ・スコープで有効になっていません。
- SPIR* カーネル関数のポインターは `!genx.kernels` メタデータに保存されるため、ESIMD のデッド引数の排除は実行できません。

- 同じフィルターを `filter_selector` に渡して返されたデバイスは等しいか比較できないことがあります。
- Windows* では、DPC++ コンパイラーは、次の場合、SYCL* ライブラリーとリンクされたアプリケーションでダイナミック C++ ランタイムの使用を強制します。

- `-fsycl` オプションを指定して `msvcrt[d].dll` とリンクする場合。
- `-fsycl` および `/MT` または `/MTd` オプションを指定してスタティック C++ ランタイムを使用してプログラムをコンパイルしようとするエラーが発生する場合。これにより、`sycl[d].dll` 境界を越える C++ オブジェクトにより引き起こされる複雑なランタイムエラーから保護されます。ただし、アプリケーションと `sycl[d].dll` 側で使用される異なるバージョンの C++ ランタイムで適切に処理されるとは限りません。

- アプリケーションをデバッグモードでコンパイルしている場合、Windows* 上で次のようなランタイム例外が発生します。この問題を回避するには、コマンドラインで `/Od` オプションを指定してコンパイルするか、IDE で **[Linker] > [General] > [Pass additional options to device compilers]** に `/Od` を追加します。

```
Unhandled exception at 0x00007FF930ED7247 (igc64.dll) in gamma-correction.exe: 0xC0000005: Access violation reading location 0x0000027455B6C000.
```

- 「[DPC++ または OpenMP* オフロードを使用してヘテロジニアス・プログラムをデバッグする場合の課題、ヒント、および既知の問題](#)」(英語) の情報も参照してください。
- FPGA がインストールされている DPC++ システムはマルチプロセス実行をサポートしていません。コンテキストを作成すると、コンテキストに関連付けられているデバイスが開き、そのプロセスのためにデバイスがロックされます。ほかのプロセスはそのデバイスを使用できません。ランタイムは実際のデバイスを照会してその情報を取得する必要があるため、`device.get_info<>()` によるデバイスに関する一部のクエリーも、デバイスを開き、そのプロセスのためにデバイスをロックします。デバイスをロックするクエリーの例を次に示します。

- `is_endian_little`
- `global_mem_size`
- `local_mem_size`
- `max_constant_buffer_size`
- `max_mem_alloc_size`
- `vendor`
- `name`
- `is_available`

- FPGA 向けにコンパイルするときに、名前なし名前空間でカーネル名を宣言すると、Summary、FMAX II Report、Area Analysis of System、Graph Viewer、Kernel Memory Viewer、Schedule Viewer などの FPGA 最適化レポートにカーネル名が正しく表示されません。この問題を回避するには、カーネル名をグローバルに宣言します。
- Windows* 上で Microsoft* Visual Studio* で FPGA エミュレーター・コードをデバッグすると、デバッガーはカーネルコードに設定されたブレークポイントで停止しません。現在、この問題の回避策はありません。
- FPGA コマンド `aocl diagnose` を使用すると、「ICD diagnostics FAILED (ICD 診断失敗)」エラーが出力されることがあります。インテル® oneAPI ベース・ツールキットを指示どおりにインストー

ルした場合、ICD も正しくインストールされているため、このエラーは無視してかまいません。oneAPI ベース・ツールキット用 Intel® FPGA アドオンパッケージが正常にインストールされていれば、このエラーが原因でコンパイルまたは FPGA ハードウェアの実行エラーが発生することはありません。

- FPGA 最適化レポートの Bottleneck Viewer が空白 (データなし) で表示されます。この問題を回避するには、Loop Analysis レポートの [Details] ペインを参照してボトルネックを特定します。
- FPGA 向けにコンパイルするときに、非常に長いカーネル名を宣言すると、コンパイルエラーになります。この問題を回避するには、カーネル名を 260 文字未満にします。
- FPGA 向けにコンパイルするときに Windows* でデバイス・コード・アーカイブを作成すると、次のリンク警告が表示される場合があります。

```
warning LNK4078: multiple '__CLANG_OFFLOAD_BUNDLE_SIZE__sync' sections
found with different attributes (40100800)
warning LNK4078: multiple '__CLANG_OFFLOAD_BUNDLE__sycl-fpg' sections
found with different attributes (40100800)
```

これらの警告は無視してかまいません。

- FPGA 固有の `-reuse-exe=<exe>` オプションは Windows* ではサポートしていません。ホストコードのみ変更するときにコンパイル時間を最小限に抑えるためにホストコードとデバイスコードを分離する方法の例は、[fast_recompile](#) (英語) FPGA チュートリアルを参照してください。
- FPGA 向けにコンパイルするときに非常に広い `struct` に読み取り専用アクセサーを使用すると、RTL のコンパイル時間 (FPGA ハードウェア・イメージ作成段階の前) が長くなることがあります。この長いコンパイル時間を回避するには、代わりに読み取り/書き込みアクセサーを使用します。
- FPGA エミュレーター向けにコンパイルされたデザインを実行すると、「OpenCL API failed (OpenCL* API 失敗)」エラーメッセージが表示されることがあります。OpenCL* API は `-5 (CL_OUT_OF_RESOURCES)` エラーを返します。この問題を回避するには、次のコマンドを使用して、エミュレーター・ランタイムで割り当てが許可されている値 (デフォルト値は 512KB) までメモリーの量を増やします。ここで、`<size>` は、整数とキロバイトの場合は KB (例えば、1024KB)、メガバイトの場合は MB (例えば、32MB) で指定します。
 - **Linux*:** `export CL_CONFIG_CPU_FORCE_PRIVATE_MEM_SIZE=<size>`
 - **Windows*:** `set CL_CONFIG_CPU_FORCE_PRIVATE_MEM_SIZE=<size>`
- Windows* で (前の問題で説明したように) FPGA エミュレーターのメモリーが不足すると、「OpenCL API failed (OpenCL* API 失敗)」エラーメッセージが表示されないことがあります。つまり、何も表示されずにエミュレーターの実行が終了します。この問題を回避するには、このメモリー不足エラーが確実に表示されるように、`try-catch` 構文を使用してカーネルコードを記述します。
- FPGA 向けにコンパイルするときに、`ivdep` 属性が適用される範囲外で宣言されたポインターが含まれていると、コンパイラーは属性を無視することがあります。次に例を示します。

```
int *p = ...
// enter new scope
{
    [[intel::ivdep(p)]]
    for (int i = 0; i < N; i++) {
        // accesses to p
    }
}
```

この例では、`ivdep` 属性を適用しても、コンパイラーはループ内の `p` へのアクセスに依存性を検出することがあります。この制限を回避するには、可能であれば、属性を適用する同じ範囲内で (`ivdep` 属性で使用することを意図した) ポインターを宣言します。この制限により正しく記述されたコードで機能エラーが発生することはありませんが、生成されたハードウェアのパフォーマンスに影響する可能性があります。

- FPGA 向けにコンパイルするときに、インテル® FPGA PAC D5005 がインストールされたシステムを使用してインテル® PAC インテル® Arria® 10 GX FPGA 搭載版をターゲットとする SYCL* アプリケーションをコンパイルすることはできません。コンパイルに成功しても、コンパイルされたバイナリーが実行時に失敗する可能性があります。現在、この問題の回避策はありません。
- インテル® PAC インテル® Arria® 10 GX FPGA 搭載版のソフトウェア・スタックとインテル® FPGA PAC D5005 のソフトウェア・スタックは、同じマシン上で互いに互換性はありません。ソフトウェア・スタックの 1 つをすでにシステムにインストールしている場合は、もう 1 つのソフトウェア・スタックをインストールする前に、`aocl uninstall` コマンドを実行してアンインストールする必要があります。
- GPU プラットフォーム (oneAPI 固有) もインストールしている場合、FPGA エミュレーターのコンパイルが失敗して、「Error: Compiler Error: OpenCL kernel compile/link FAILED (エラー: コンパイルエラー: OpenCL* カーネルのコンパイル/リンクに失敗)」エラーメッセージが表示されることがあります。このエラーを回避して FPGA エミュレーターのコンパイルを正常に行うには、次のいずれかのソリューションを実行します。
 - **ソリューション 1:** エミュレーション・フローをターゲットにするときに、すべての `dpcpp` コマンドに `-Xsfast-emulator` オプションを追加します。
 - **ソリューション 2:** 次にリストされている OS 固有のコマンドの 1 つを実行します。
 - **Ubuntu* 18.04:** `export LD_PRELOAD=/usr/lib/x86_64-linux-gnu/libstdc++.so.6`
 - **Ubuntu* 20.04:** `export LD_PRELOAD=/lib/x86_64-linux-gnu/libstdc++.so.6`
 - **SLES 15:** `export LD_PRELOAD=/usr/lib64/libstdc++.so.6`
 - **CentOS* 8.x:** `export LD_PRELOAD=/lib64/libstdc++.so.6`
 - **RHEL 8.1:** `export LD_PRELOAD=/lib64/libstdc++.so.6`

注: コードサンプルの CMake ファイルの `dpcpp` コマンドを変更するのは面倒なため、コードサンプルを利用する場合はソリューション 2 を推奨します。

終了予定のサポート

`-mkl` コンパイラー・オプションを `-qmk1` に置換

Linux* の `-mkl` コンパイラー・オプションは古いオプション (非推奨) で、将来のリリースで削除される予定です。将来のリリースでは、`-qmk1` に置換されます。このコンパイラー・オプションは、インテル® oneAPI マス・カーネル・ライブラリー (インテル® oneMKL) の特定のライブラリーにリンクするようにコンパイラーに指示します。

法務上の注意書き

インテルのテクノロジーを使用するには、対応したハードウェア、ソフトウェア、またはサービスの有効化が必要となる場合があります。

絶対的なセキュリティを提供できる製品またはコンポーネントはありません。

実際の費用と結果は異なる場合があります。

© Intel Corporation. Intel、インテル、Intel ロゴ、Arria 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。

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依存の最適化は、インテル® マイクロプロセッサ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。

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。

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。

インテルは、明示されているか否かにかかわらず、いかなる保証もいたしません。ここにいう保証には、商品適格性、特定目的への適合性、および非侵害性の黙示の保証、ならびに履行の過程、取引の過程、または取引での使用から生じるあらゆる保証を含みますが、これらに限定されるわけではありません。

製品とパフォーマンス情報

¹ 性能は、使用法、構成、およびその他の要因によって異なります。

詳細については、www.Intel.com/PerformanceIndex (英語) を参照してください。