インテル® DPC++ 互換性ツール (インテル® DPCT) 導入ガイド

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「Get Started with the Intel® DPC++ Compatibility Tool」の日本語参考訳です。原文は更新される可能性があります。原文と翻訳文の内容が異なる場合は原文を優先してください。

この記事の PDF 版はこちらからご利用になれます。

作成日: 2023 年 7 月 13 日
バージョン: 2023.2

インテル® DPC++ 互換性ツール (インテル® DPCT) は、既存の CUDA* コードから SYCL* コードへの移行を支援します。CUDA* 言語カーネルとライブラリー API 呼び出しを移植し、CUDA* コードの 90% ~ 95% を SYCL* コードに移行します。さらに、移行の完了とコードの調整に役立つインライン警告を挿入します。

インテル® DPCT は、CUDA* バージョン 8.0、9.x、10.x、11.x、12.0 ~ 12.1 で実装されたプログラムの移行をサポートします。サポートされる言語とバージョンのリストは将来拡張される可能性があります。

- インテル® DPCT の詳細は、『インテル® DPC++ 互換性ツール (インテル® DPCT) 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。
- 最新情報および既知の問題に関する情報については、「リリースノート」(英語) を参照してください。

準備

1. インテル® DPCT をインストールします。

   インテル® DPCT は、インテル® oneAPI ベース・ツールキットに含まれています。まだインテル® oneAPI ベース・ツールキットをインストールしていない場合は、「インストール・ガイド」(英語) の手順に従ってください。

   インテル® DPCT は、スタンドアロン製品としてダウンロード (英語) することもできます。

2. CUDA* ヘッダーがツールからアクセス可能であることを確認します。

   特定の CUDA* ヘッダーファイル (プロジェクト固有) は、インテル® DPCT からアクセスできる必要があります。インテル® DPCT は、次のデフォルトの場所にある CUDA* ヘッダーファイルを探します。

   - `/usr/local/cuda/include`
   - `/usr/local/cuda-x.y/include` (`x.y` は、8.0、9.x、10.x、11.x、12.0-12.1 のいずれか)

注:
NVIDIA ではないソースから CUDA* をインストールした場合、ヘッダーファイルは別の場所にある可能性があります。ヘッダーファイルの場所を特定し、ツールの実行時に `--cuda-include-path=<path/to/cuda/include>` オプションを使用して場所を指定します。
CUDA* インクルード・パスは、(移行する必要がある) ソースコードが配置されているディレクトリーと同じ、またはその子パスであってはなりません。

3. インテル® DPCT で移行したコードで使用される DPC++ 固有の拡張をサポートするコンパイラーをインストールします。
   - インテル® oneAPI DPC++/C++ コンパイラー
   - oneAPI DPC++ コンパイラー (英語)

4. setvars スクリプト (英語) を使用して環境変数を設定します。

5. (オプション) プログラムが GPU をターゲットにしている場合は、適切な GPU ドライバーまたはプラグインをインストールして、プログラムをインテル、AMD*、または NVIDIA* GPU 上で実行できるようにコンパイルします。
   - インテル® GPU を使用する場合は、最新のインテル® GPU ドライバーをインストールします (英語)。
   - AMD* GPU を使用する場合は、oneAPI for AMD* GPU プラグインをインストールします。
   - NVIDIA* GPU を使用する場合は、oneAPI for NVIDIA* GPU プラグインをインストールします。

インテル® DPCT の実行

インテル® DPCT をコマンドラインから実行し、コマンドライン・オプションを使用して移行の設定を指定できます。一般的なコマンド構文を以下に示します。

```
dpct &lbrack;options&rbrack; &lbrack;<source0>... <sourceN>&rbrack;
```

注: `c2s` は `dpct` コマンドのエイリアスであり、代わりに使用できます。

インテル® DPCT 固有オプションの一覧を表示するには、`--help` を使用します。

```
dpct --help
```

言語パーサー (clang) オプションの一覧を表示するには、clang オプションとして `-help` を渡します。

```
dpct -- -help
```

コマンドライン・オプションの完全なリストは、「コマンドライン・オプションのリファレンス」を参照してください。

移行するファイルの指定

移行するディレクトリーまたはファイルを指定しない場合、インテル® DPCT は現在のディレクトリーにあるソースファイルの移行を試みます。デフォルトの出力ディレクトリーは `dpct_output` です。出力ディレクトリーを指定するには `--out-root` オプションを使用します。

オプションで、移行するソースファイルのパスを指定できます。パスはコンパイル・データベースで見つけることができます。以下の例は、移行するファイルまたはディレクトリーを指定する方法を示します。

- 単一のソースファイルを移行します。

  ```
  dpct source.cpp
  ```

- コンパイル・データベースにあるすべてのファイルを移行します。

  ```
  dpct -p=<path to the location of compilation database file>
  ```

- コンパイル・データベースにある 1 つのファイルを移行します。

  ```
  dpct -p=<path to the location of compilation database file> source.cpp
  ```

- `--in-root` オプションで指定したディレクトリーのソースファイルを移行し、生成したファイルを `--out-root` オプションで指定したディレクトリーに配置します。

  ```
  dpct --in-root=foo --out-root=bar
  ```

警告を理解する

ファイルの移行中、インテル® DPCT は、SYCL* に準拠するため、または正しいコードにするため注意が必要な箇所を特定します。

そして、生成するソースファイルに警告を挿入し、出力に表示します。以下に例を示します。

```
/path/to/file.hpp:26:1: warning: DPCT1003:0: Migrated API does not return error
code. (*,0) is inserted. You may need to rewrite this code.
// source code line for which warning was generated
^
```

特定の警告の意味については、「診断リファレンス」を参照してください。

サンプルコードの入手

インテル® DPCT のサンプルコードを使用して、移行プロセスやツールの機能に慣れてください。

サンプルコードは、以下の方法で入手できます。

- oneAPI CLI Samples Browser を使用して、Intel® DPC++ Compatibility Tool カテゴリーからサンプルを選択します。oneAPI CLI Samples Browser を使用したサンプルのダウンロード、および IDE を使用したサンプルの実行の詳細については、以下を参照してください。
  - インテル® oneAPI ベース・ツールキット (Windows* 版) 導入ガイド (英語)
  - インテル® oneAPI ベース・ツールキット (Linux* 版) 導入ガイド (英語)

- GitHub* (英語) からサンプルをダウンロードします。

各サンプルの README に詳しい移行手順があります。

サンプル・プロジェクト	説明
Vector Add • `vector_add.cu`	Vector Add サンプルは、単純なプログラムを CUDA* から SYCL* に移行する方法を示します。このサンプルを使用して、インテル® DPCT を使用するため開発環境が正しく設定されているかどうかを確認できます。
Folder Options • `main.cu` • `bar/util.cu` • `bar/util.h`	Folder Options サンプルは、より複雑なプロジェクトの移行とツールオプションの使用方法を示します。
Rodinia needleman-wunsch • `needle.cu` • `needle.h` • `needle_kernel.cu`	Rodinia needleman-wunsch サンプルは、Make/CMake* プロジェクトを CUDA* から SYCL* に移行する方法を示します。

Vector Add サンプルの移行

Vector Add サンプルは、単純な CUDA* プログラムを SYCL* 準拠のコードに移行する方法を示します。この単純なプログラムは、[1..N] の 2 つのベクトルを加算し、結果を表示します。このプログラムは CPU 向けです。

次の手順に従って、インテル® DPCT を使用して Vector Add サンプルを移行します。

1. Vector Add サンプル (英語) をダウンロードします。
2. Vector Add サンプルのルートに移動します。このサンプルでは、`src` フォルダーに 1 つの CUDA* ファイル (`vector_add.cu`) があります。
3. サンプル・プロジェクトのルートフォルダーから、インテル® DPCT を実行します。

   ```
   dpct --in-root=. src/vector_add.cu
   ```

   `--in-root` オプションは、移行するプログラムソースのルートを指定します。インテル® DPCT は、`--in-root` ディレクトリー内のファイルとフォルダーのみを移行対象と見なします。`--in-root` ディレクトリー外のファイルは、`--in-root` ディレクトリー内のソースファイルでインクルードされていても移行されません。デフォルトでは、移行されたファイルは `dpct_output` という名前の新しいフォルダーに作成されます。

   移行コマンドを実行すると、出力フォルダーに新しい SYCL* ソースファイルが作成されます。

   ```
   dpct_output
   └── src
       └── vector_add.dp.cpp
   ```

   移行されたファイルの相対パスは維持されます。

4. 移行されたソースコードを検査し、インテル® DPCT によって生成された DPCT 警告に対処してください。

   このサンプルでは、以下の警告が生成されます。

   ```
   warning: DPCT1003:0: Migrated API does not return error code. (*, 0) is
   inserted. You may need to rewrite this code.
   ```

   「診断リファレンス」の「DPCT1003」を確認します。

   この警告は、SYCL* がエラーコードの代わりに、例外を使ってエラーを報告する場合に出力されます。このサンプルでは、インテル® DPCT は、失敗時に終了する条件文を削除し、代わりにコードを try ブロックで囲んでいます。元のコードのエラーステータス変数は残して、常にエラーコード 0 を代入するようにソースを変更しています。

   警告の説明では、修正方法が提案されています。このサンプルでは、ステータス変数は不要であるため、手動で削除してください。

5. 移行されたコードをコンパイルします。

   ```
   icpx -fsycl -I<install_dir>/include src/vector_add.dp.cpp
   ```

   ここで、<install_dir> はインテル® DPCT のインストール・ディレクトリーです。

6. 移行されたプログラムを実行します。

   注: Vector Add サンプルは CPU 向けです。ONEAPI_DEVICE_SELECTOR 環境変数を使用して、CPU をターゲットにしてください。

   ```
   ONEAPI_DEVICE_SELECTOR=*:cpu
   ./vector_add
   ```

   2 つのベクトルを加算した結果 (`[1..N] + [1..N]`) を示す偶数のブロックが出力されるはずです。

より複雑なサンプルの移行手順については、『インテル® DPC++ 互換性ツール・デベロッパー・ガイドおよびリファレンス』の「プロジェクトの移行」セクションを参照してください。

関連情報

目次	説明
インテル® DPC++ 互換性ツール・デベロッパー・ガイドおよびリファレンス	インテル® DPCT の機能、ワークフロー、使用法について詳しく説明しています。
オンデマンド・ウェビナー: 既存の CUDA* コードから DPC++ コードへの移行 (英語)	カーネルと API 呼び出しの両方を一度に移行できるインテル® DPCT を使用して、CUDA* コードをデータ並列 C++ (DPC++) (英語) に移行する方法を説明します。
インテル® oneAPI ツールキットのインストール・ガイド (英語)	さまざまなインストーラー・モードおよびパッケージ・マネージャーを使用したインテル® oneAPI パッケージの入手方法とインストール方法について詳しく説明します。
SYCL* 仕様バージョン 1.2.1 PDF (英語)	SYCL* 仕様の PDF です。SYCL* が OpenCL* と最新の C++ をどのように統合するかを説明します。
SYCL* 2020 仕様	SYCL* 2020 仕様の日本語版 PDF です。
Khronos* SYCL* の概要 (英語)	Khronos Group が提供する SYCL* の概要です。
clang による CUDA* のコンパイル (英語)	clang の CUDA* サポートについての説明です。
インテルの LLVM SYCL* 拡張 (英語)	SYCL* 仕様に対する拡張の提案です。
Yocto* Project 向けのレイヤー (英語)	meta-intel レイヤーを使用して Yocto* Project のビルドに oneAPI コンポーネントを追加できます。

* SYCL は Khronos Group, Inc. の登録商標です。