インテル® C++ コンパイラーから新しい C++ コンパイラーへの移行

icc から icx/icpx に切り替え

エクセルソフト株式会社

ご紹介内容

資料の一部に下記の意味で使用することがあります
icc - インテル® C++ コンパイラー・クラシック
icx - インテル® DPC++/C++ コンパイラー

- インテル® C++ コンパイラー・クラシックから
  インテル® DPC++/C++ コンパイラーへ切り替えましょう
  次期メジャーアップデートのタイミングで icc の削除が予定されています
- ポーティング・ガイドをもとに移行にあたって
  icc からの主な変更点をご案内します
  - Porting Guide for ICC Users to DPCPP or ICX (英語)
  - 日本語参考訳 - インテル® oneAPI ポーティング・ガイド
    ICX へ移行する ICC ユーザー向け

インテル® oneAPI DPC++/C++ コンパイラー

- LLVM をベースに新規開発されたコンパイラー
- Clang フロントエンドを採用しつつ
  インテルによる独自の最適化およびコード生成を実装
  clang のコンパイラーオプションを受け付けます
- SYCL* 2020 および OpenMP* 5.0 のオフロード機能をサポート
  FPGA、GPU ターゲットを含むコード実装
- icc の後継として移行が推奨されています
  ただし icc と異なるコンパイラーなので計算結果の
  違いや最適化への影響を考慮する必要があります

基本オプション

いままで	Linux* コマンド	Windows* コマンド
icc/icl	**icx**	icx
icpc/icl	**icpx**	icpx

- icx/icpx コマンド

```
> icx sample.c
> icpx sample.cpp
```

- インテル® MPI ライブラリーから呼び出す
  -cc、-cxx オプションへ icx/ipcx を指定

```
> mpicc -cc=icx sample_mpi.c
> mpiicpc -cxx=icpx sample_mpi.cpp
```

- 基本はコンパイラーのコマンドを切り替えるだけ icc → icx or icpx
- Visual Studio* プロジェクトも使用するコンパイラーセットを切り替え

コンパイラーオプションの方針

- 多くのオプションは利用可能な一方で
  一部のオプションは変更や削除されています

  引き続きサポートされるオプションと icx で削除されたオプションは
  -qnextgen-diag、/Qnextgen-diag オプションの指定により取得可能
- icc との互換性を目的に一部で Clang のデフォルトと
  異なるオプションが設定されています

  例:　icx: -fp-model=fast
  　　clang: -fp-model=precise

ベクトル化機能

- LLVM が提供するベクトライザーとは別に
  インテル製CPU、GPU 向けに設計
  icc が使用するベクトル化機能とは異なるアプローチを採用します

- -x 及び -ax は引き続き動作します
  icx では -O2 および -O3 と -x の併用を推奨
  ループとベクトル化の最適化に大きく影響

```
> icpx -O2 -xcore-avx2 sample.cpp
> icpx -O2 -mavx2 sample.cpp
```

IPO、PGO

- icc にて提供されていた IPO、PGO は
  LLVM が提供する LTO、PGO に置き換え

  LTO - Link Time Optimization

  PGO - Profile-Guided Optimizations
- icc の実装と比較して LTO、PGO は異なる手法で実装

IPO

- LLVM が提供する liblto をリンカーが使用
  icc のリンカーツール xild, xilink は使用されません
- -flto, -flto=[arg]
  arg = full (デフォルト)もしくは thin
  ThinLTO は full と比較してメモリ消費量を抑えつつビルド時間の短縮を目指した異なる LTO の手続きを提供します

```
> icx -O2 -xcore-avx2 -flto *.c
> icx -O2 -xcore-avx2 -flto=thin *.c
```

- オプションの指定を変更 (今のところ任意)
  -ipo 及び /Qipo から -flto へエイリアスされています
  -ipo, /Qipo → -flto オプションの利用

PGO

- icc の PGO とは全く異なる手続き
- -fprofile-instr-generate、-fprofile-instr-use
- インストルメント方式と外部プロファイラーによるサンプリング
- オプションの指定を変更

  -prof-gen,/Qprof-gen → -fprofile-instr-generate
  -prof-use, /Qprof-use → -fprofile-instr-use
- llvm-profdata ツールを別途ダウンロード

  llvm パッケージに付属

インストルメント方式の PGO 手順

1. -fprofile-instr-generate を追加してコンパイル

```
> icx -O2 -xcore-avx2 -fprofile-instr-generate *.c
```

2. 必要なコードパスを通るように実行

```
> ./a.out
```

3. llvm-profdata ツールに生成されたデータを投入

```
llvm-profdata-14 merge --output=default.profdata ./default.profraw
```

4. -fprofile-instr-use=<ファイル名> を追加して再コンパイル

```
icx -fprofile-instr-use=default.profdata *.c
```

浮動小数点演算の再現性

- -fp-model <type>, /fp:<type>
  <type>= precise, fast(デフォルト), strict

```
> icx -O2 -xcore-avx2 -fp-model precise *.c
> icx /O2 /Qxcore-avx2 /fp:strict *.c
```

- consistent, source, double, extendedは利用できません
  -fp-model consistent → -fp-model precise -fimf-arch-consistency=true -no-fma

プラグマのサポート

icc で採用されていたメッセージ番号による表記は削除され
Clang/LLVM ベース のメッセージを出力します
-diag, /Qdiag から始まるメッセージ出力を制御するオプションは機能しません

- icx では一部の実装を削除しており icc や gcc の
  プラグマを認識しない場合があります

  icx - Intel-Specific Pragma Reference

  icc - Intel-Specific Pragma Reference
- -Wunknown-pragmas で警告を表示

```
#pragma unroll
#pragma ivdep
#pragma parallel
#pragma vector
     for (i=1;i<n;i++)
     {
        c[i] = a[i] + b[i];
     }
```

```
> icx -Wunknown-pragmas vec_add.c
Intel(R) oneAPI DPC++/C++ Compiler for applications running on Intel(R)
64, Version 2023.2.0 Build 20230627
Copyright (C) 1985-2023 Intel Corporation. All rights reserved.

vec_add.c(89,9): warning: unknown pragma ignored [-Wunknown-pragmas]
#pragma parallel
        ^
1 warning generated.
```

OpenMP* オプション

- -fiopenmp を利用するように変更
  インテルの OpenMP* 実装を使用
- -fopenmp は非推奨 → 削除予定
  LLVM の OpenMP* 実装を使用
  デバッグ目的での利用を想定
- OpenMP* の target ディレクティブを使用したコードは -fopenmp-targets=spir64 を追加

```
icx -fiopenmp -fopenmp-targets=spir64
icpx -fiopenmp -fopenmp-targets=spir64
```

最適化レポート

- LLVM ベースの YAML ファイルへの出力
  + 従来のテキストファイルのレポート出力
- -qopt-report=<n>, /Qopt-report:<n>
  <n> = 1~3, デフォルトは 2

```
> icx -O2 -xcore-avx2 -qopt-report -qopt-report-stdout *.c
> icx -O2 -xcore-avx2 -qopt-report=3 *.c
```

- *.yaml と *.optrpt ファイルから最適化レポートの参照
  YAML ファイルを確認する場合は別途 llvm-opt-report ツールを利用(要ダウンロード)

まとめ

- icx は Clang/LLVM をベースに新規開発されたコンパイラーです
- icx はコンパイルコマンドを切り替えるだけ利用できます
- icc とは異なるオプションや同じオプションでも実装が異なります
  ベクトル化機能の利用にはオプションを必ず追加

お問い合わせはこちらまで
https://www.xlsoft.com/jp/qa