

インテル® C/C++ コンパイラーが LLVM への対応を完了

この記事は、software.intel.com に公開されている「[Intel C/C++ compilers complete adoption of LLVM](#)」の日本語参考訳です。

新しい世代のインテル® C/C++ コンパイラーは、LLVM オープンソース・インフラストラクチャーを採用しているため、従来のコンパイラーよりも優れています。

LLVM は、インテル® アーキテクチャー向けに最高の C/C++ コンパイラーを提供するというインテルの目標にも一致しています。LLVM を採用した最新のインテル® C/C++ コンパイラーは、コンパイル時間の短縮、最適化の向上、標準サポートの強化、GPU や FPGA へのオフロードのサポートを実現します。

この記事では、インテル® コンパイラーの LLVM への対応に関する情報を共有します。インテル® C/C++ コンパイラーが LLVM を採用したことの意味、採用した理由、そして今後の展望についてお話しします。さらに、(まだ完了していない) インテル® Fortran コンパイラーの LLVM への対応に関する最新情報も提供します。

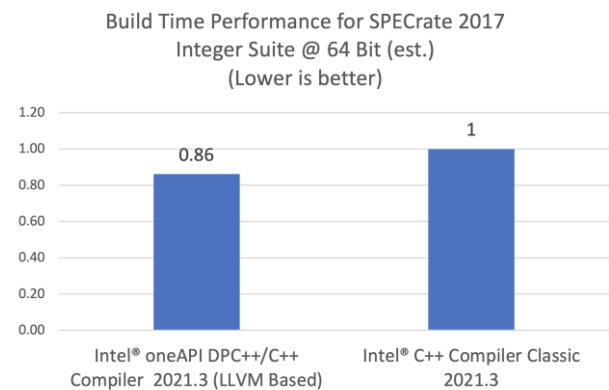
LLVM を採用することで得られるメリットは数多くあります。インテルは、従来のコンパイラーから LLVM ベースのコンパイラーにアップグレードする際のアドバイスを提供し、インテル® コンパイラーを使用している開発者に多くのメリットをもたらしつつ、可能な限りシームレスなアップグレードを実現することをお約束します。

LLVM を採用することで得られるメリット

LLVM オープンソース・プロジェクトは、複数のプロセッサ・アーキテクチャーとプログラミング言語をサポートする、モジュール化された再利用可能なコンパイラーとツールチェーン技術のコレクションです。Clang オープンソース・プロジェクトは、LLVM プロジェクトの最新の言語標準をサポートする C/C++ フロントエンドを提供します。LLVM と Clang は、大規模で非常に活発な開発コミュニティによって維持されています。

LLVM を採用することで得られるメリットはたくさんありますが、まずは**ビルド時間の短縮**から見ていきましょう。Clang は高速です。これはありがたいことです。[インテル® oneAPI ツールキット 2021.3](#) に含まれるインテル® C/C++ コンパイラーを使用したところ、ビルド時間が 14% 短縮しました。さらに、Clang を採用することで、最新の C++ 言語標準をサポートするコミュニティの取り組みに貢献し、その恩恵を受けることができました。

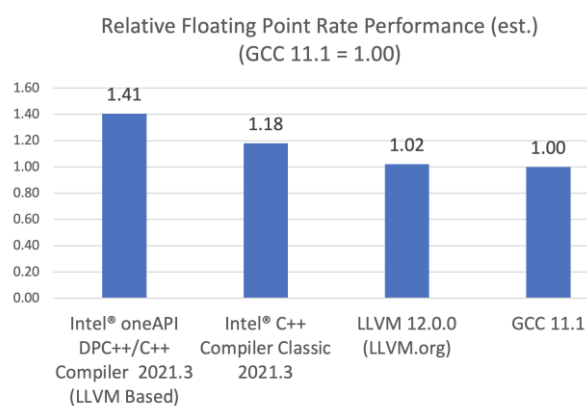
インテルは、長年にわたってオープンソース・プロジェクトへの貢献とサポートに取り組んでおり、その中には LLVM への 10 年に及ぶ貢献も含まれています。現在、インテルは最適化レポートの追加、浮動小数点モデルのサポートの拡張、ベクトル化の強化などにおいて、積極的に協力しています。インテルは LLVM プロジェクトに直接貢献していますが、SYCL* サポート用のステー징・エリア ([インテルの LLVM テクノロジー・プロジェクト](#) (英語)) も用意しています。



Estimated: Built time measurement of the geometric mean of the C/C++ workloads from the SPECrate* 2017 Integer suite

インテル® C/C++ コンパイラーのほうが、インテル® アーキテクチャー用のベース Clang + LLVM コンパイラーよりも高いパフォーマンスが期待できます。インテル® C/C++ コンパイラーのデフォルトは、LLVM オープンソース・インフラストラクチャーを採用したバージョン (icx) です。インテルは、Clang や LLVM の最適化を含め、Clang および LLVM プロジェクトに貢献してきました。最適化手法によっては斬新すぎたり、インテル® アーキテクチャーに特化しているため、すべての最適化手法が標準に組みこまれるわけではありません。これは当然のことで、LLVM を採用しているほかのコンパイラーでも同様です。

この記事の一連のパフォーマンス測定には、インテル® oneAPI ツールキット 2021.3 とともにリリースされた最新のインテル® C/C++ コンパイラーを使用しました。インテル® アーキテクチャー向けの主要な C/C++ コンパイラーになるという目標と一致して、測定結果は、インテル® C/C++ コンパイラーがほかの選択肢よりも優れていることを示しています。LLVM ベースの新しいインテル® C/C++ コンパイラーは、インテル® C/C++ コンパイラー・クラシックと同等以上のパフォーマンスを発揮します。今こそ、コンパイラーをアップグレードするときです。ここではその一例を紹介します。記事の最後に追加のベンチマークを掲載しています。



Estimated: internal measurement of the geometric mean of the C/C++ workloads from the SPECrate 2017 Floating Point suite

インテル® C/C++ コンパイラーは、これまでもパフォーマンスにおいてリーダーシップを示してきました。ここでは gcc と比較して、インテル® C/C++ コンパイラー・クラシックは **18% 優れており** (英語)、LLVM ベースのインテル® C/C++ コンパイラーは **41% 優れています** (英語)。

インテルの進化するプラットフォームをサポートするため、LLVM ベースのコンパイラーの新機能とハードウェアのサポートに重点を置いて、業界をリードする CPU の最適化を提供するという継続的な取り組みに加えて、GPU と FPGA 向けの高度な最適化をサポートしました。LLVM ベースのコンパイラーでは、SYCL*、C++20、OpenMP* 5.1、OpenMP* の GPU ターゲットデバイスをサポートしています。

ユーザーの皆様には、ビルド時間の短縮、より高いレベルの最適化、新しい機能を利用できる、LLVM ベースの C/C++ コンパイラーに移行することをお勧めします。インテルは、継続的なイノベーションと業界をリードする最適化の飽くなき追求を支援するため、LLVM に長期的に取り組んでいます。

インテル® Parallel Studio XE のコンパイラーについて

2007 年にインテルは、並列処理のサポートを強調するため、ツールの名称を「インテル® Parallel Studio XE」に変更しました。当時、マルチコアプロセッサにより、並列プログラミングがユビキタスとなる世界の変化がありました。それは、デュアルコア・プロセッサがシングルコア・プロセッサに取って代わることから始まりました。現在、コア数は数十に達しており、依然として増加傾向にあります。



ホモジニアス・システムの並列プログラミングがユビキタスになったように、ヘテロジニアス・システムの並列プログラミングも同様にユビキタスへの道を進んでいると考えられます。マルチコアの並列処理とは異なり、ヘテロジニアス・プログラミングは、複数のベンダーの計算能力にまたがります。コンパイラー、ライブラリー、

フレームワーク、そしてすべてのソフトウェア開発ツールにおいて、マルチベンダーのオープンなアプローチをサポートするため業界全体で取り組まなければ、プログラミングの細分化につながる恐れがあります。

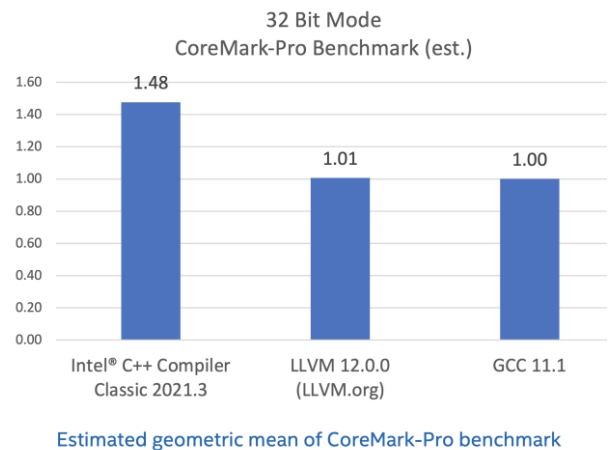
ヘテロジニアス並列処理に対する oneAPI のオープンなアプローチを強調するため、インテルは次世代のツールにこの名前を付けました。これらは、数十年にわたって業界で信頼されてきた製品品質のツールであり、[oneAPI 仕様](#) (英語) と [SYCL* 標準](#) (英語) を採用することで、ヘテロジニアス・プログラミングをサポートするように拡張されています。無料でダウンロードして、すぐにツールを使用できます。[インテルのコミュニティー・フォーラム](#) (英語) からコミュニティーによるサポートを利用できます。また、インテルのエンジニアに質問や問題、その他の技術サポートを問い合わせ可能な[優先サポート](#) (英語) を利用できる有償版も用意されています。

C/C++ は対応を完了

新しいプロジェクトは、LLVM ベースのインテル® C/C++ コンパイラーで開始し、既存のプロジェクトは、今年中に新しいコンパイラーに移行する計画を立てることを推奨します。将来的には、インテル® C/C++ コンパイラー・クラシックは、「レガシー製品サポート」モードに移行し、定期的なアップデートの提供が終了し、インテル® oneAPI ツールキットに含まれなくなります。

新しい LLVM ベースのインテル® C/C++ コンパイラーは、インテル® C/C++ コンパイラー・クラシックと同等のパフォーマンスだけでなく、インテルの最高の最適化テクノロジーを提供します。今すぐ新しいインテル® C/C++ コンパイラーを試してその効果を実感し、フィードバックをお送りください。

インテル® C/C++ コンパイラー・クラシックから LLVM ベースのインテル® C/C++ コンパイラーへの移行を支援する[優れたガイド](#)が用意されています。まず気付くのは、コンパイラーの名前が icx に変わっていることです。これにより、インテル® C/C++ コンパイラー・クラシックと LLVM ベースの新しいインテル® C/C++ コンパイラーの両方をインストールして、コンパイル時に選択できます。すでに多くのユーザーが、LLVM ベースのインテル® C/C++ コンパイラーのみを使用するように移行を完了しています。既知の問題や制限事項の詳細は、[最新のリリースノート](#) (英語) を参照してください ([インテル® C/C++ コンパイラー・クラシックのリリースノート](#) (英語) もあります)。コンパイラー・エキスパートの話をライブで聞いたり、録画セッションをオンデマンドで視聴可能な「[Talk to the Expert](#)」ウェビナーもぜひご覧ください。



LLVM ベースのインテル® Fortran コンパイラーへの対応

インテル® Fortran は、長年にわたって、広範な標準サポートと優れたパフォーマンスを提供してきました。この伝統は、ベータプログラムが完了した後、LLVM ベースのインテル® Fortran コンパイラーに引き継がれます。皆様からのフィードバックをお待ちしております。

LLVM ベースのベータ版インテル® Fortran コンパイラーでは、Fortran 標準を幅広くサポートしていますが、一部の機能はまだ実装されていません。リリースごとの各機能のサポート状況は、「[ベータ版インテル® Fortran コンパイラーの Fortran 言語と OpenMP* 機能](#)」を参照してください。インテル® Fortran コンパイラーの[リリースノート](#)には、インテル® Fortran コンパイラー・クラシックと LLVM ベースのベータ版インテル® Fortran コンパイラーの両方の情報が記載されています。

インテル® Fortran コンパイラーの LLVM 対応状況については、2021 年後半に最新情報をお届けする予定です。

インテル® コンパイラーにとって新たな素晴らしい章の始まり

インテル® C/C++ および Fortran コンパイラーは、1990 年代初頭の UNIX* System V コンパイラーから始まり、1990 年代半ばには Multiflow 社のコンパイラー・テクノロジーが追加され、2000 年代には DEC/Compaq の Fortran チームと Kuck and Associates Inc. (KAI) の OpenMP* および並列処理に関する専門知識が追加されました。40 年目を迎えたインテル® コンパイラーは、LLVM コンパイラー・テクノロジーとともにその歩みを続けています。インテル® コンパイラーは、強力な標準サポート、信頼性の高いコードの最適化、そしてお客様のニーズをサポートするための強力な取り組みを続けています。さらに、ヘテロジニアス・プログラミングのサポートにおいても先導的な役割を果たしています。

インテルは、インテル® C/C++ および Fortran コンパイラーが、世界を変えるアプリケーションを構築する重要かつ有用なツールとなるよう、引き続き尽力してまいります。

「Talk to the Expert」ウェビナー

インテルでは、インテル® コンパイラー・テクノロジーのエキスパートによるインタラクティブなライブセッションを提供しています。「[Talk to the Expert](#)」(英語) のライブセッションでは、その場で質問をして回答を得ることができます。セッションの録画は、後日 [オンデマンド](#) (英語) で利用できます。また、[コミュニティ・フォーラム](#) (英語) では、いつでも質問をすることができます。

「[Talk to the Experts](#)」(英語) ページで登録すると、セッションの参加方法に関する詳細を確認したり、変更があった場合に通知を受け取ることができます。

次の 2 つのセッションをお勧めします。

- [2021 年 8 月 19 日 \(米国太平洋時間\): インテル® oneAPI 2021.3: 最新の LLVM ベースのコンパイラーの導入](#) (英語)
このブログで紹介した内容についてさらに詳しく説明します。
- [2021 年 9 月 9 日 \(米国太平洋時間\): コンパイラー・ドライバー、プラグマ、組み込み関数](#) (英語)
インテル® C/C++ コンパイラー・クラシックから LLVM ベースのインテル® C/C++ コンパイラーへの移行を支援する [優れたガイド](#) の内容を取り上げます。新しいコンパイラーでは、LLVM コンパイラー・テクノロジーに忠実であることを選択したため、ドライバーにはいくつかの違いがあり、プラグマや組み込み関数にも違いがあります。このセッションでは、これらについて詳しく説明します。難しいことではありませんが、エキスパートの話聞くことは有用であり、必要に応じて質問することもできます。

最新のインテル® コンパイラーを無料でダウンロード可能

インテル® コンパイラーを使用することで、インテル® アーキテクチャーと OpenMP* の最適化に関するインテルの長年にわたる経験と LLVM の両方の利点が得られます。

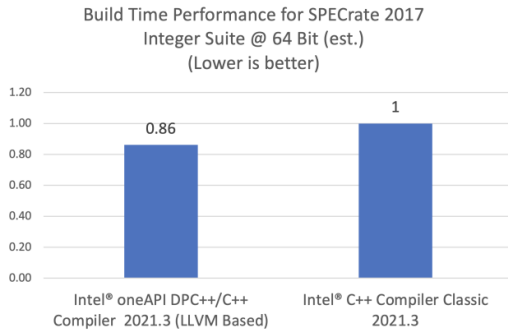
[インテル® oneAPI ツールキットのウェブサイト](#) (英語) からダウンロード可能

ご意見、ご感想、ご提案などは、[community.intel.com](#) の [James-Reinders-Blog](#) (英語) にコメントをお寄せください。

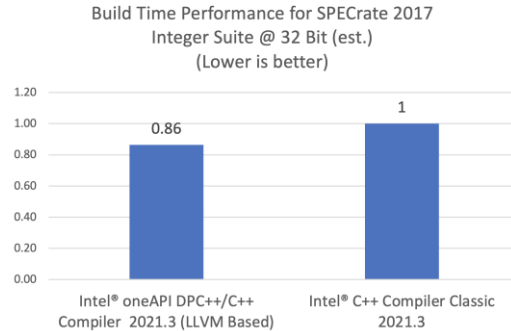
その他のベンチマークとシステム構成の詳細

これらのベンチマークは、LLVM ベースのコンパイラーがすべてのユーザーに推奨できるコンパイラーになったことを示しています。

コンパイル時間の短縮



Estimated: Built time measurement of the geometric mean of the C/C++ workloads from the SPECrate* 2017 Integer suite



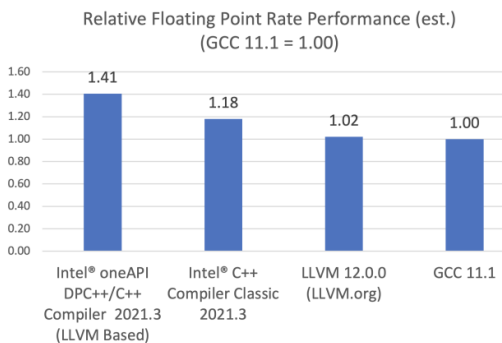
Estimated: Built time measurement of the geometric mean of the C/C++ workloads from the SPECrate* 2017 Integer suite

SPEC CPU* 2017 ベンチマーク・パッケージには、システムのプロセッサ、メモリー・サブシステム、コンパイラーに負荷をかけ、計算集約型のパフォーマンスを測定して比較する、業界標準の CPU 集約型のスイートが含まれています。SPEC ベンチマークの詳細は、<https://www.spec.org> (英語) を参照してください。

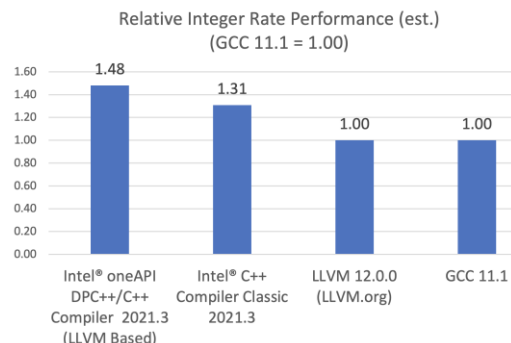
Configuration: Testing by Intel as of Jun 10,2021. Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz, 16G x2 DDR4 2666. Red Hat Enterprise Linux release 8.0 (Ootpa), 4.18.0-80.el8.x86_64. Software: Intel(R) 64 Compiler Classic for applications running on Intel(R) 64, Version 2021.3.0 Build 20210604_000000. Intel(R) oneAPI DPC++/C++ Compiler for applications running on Intel(R) 64, Version 2021.3.0 Build 20210604. Compiler switches: Intel(R) 64 Compiler Classic: -O2 -xCORE-AVX512, Intel(R) oneAPI DPC++/C++ Compiler: -O2 -xCORE-AVX512

最適化されたパフォーマンス

SPECrate* 2017 (推定値)



Estimated: internal measurement of the geometric mean of the C/C++ workloads from the SPECrate 2017 Floating Point suite

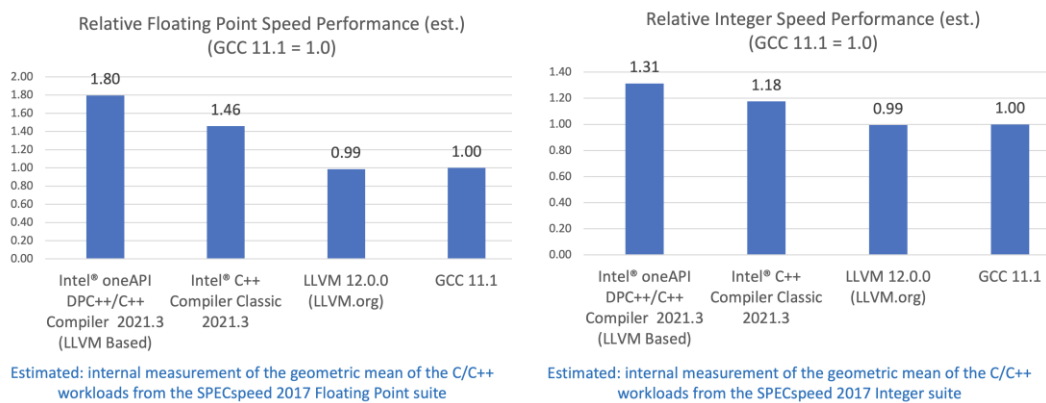


Estimated: internal measurement of the geometric mean of the C/C++ workloads from the SPECrate 2017 Integer suite

SPEC CPU* 2017 ベンチマーク・パッケージには、システムのプロセッサ、メモリー・サブシステム、コンパイラに負荷をかけ、計算集約型のパフォーマンスを測定して比較する、業界標準の CPU 集約型のスイートが含まれています。SPEC ベンチマークの詳細は、<https://www.spec.org> (英語) を参照してください。

Configuration: Testing by Intel as of Jun 10, 2021. Intel(R) Xeon(R) Platinum 8380 CPU @ 2.30GHz, 2 socket, Hyper Thread on, Turbo on, 32G x16 DDR4 3200 (1DPC). Red Hat Enterprise Linux release 8.2 (Ootpa), 4.18.0-193.el8.x86_64. Software: Intel(R) oneAPI DPC++/C++ Compiler for applications running on Intel(R) 64, Version 2021.3.0 Build 20210604. Intel(R) C++ Intel(R) 64 Compiler Classic for applications running on Intel(R) 64, Version 2021.3.0 Build 20210604_000000, GCC 11.1, Clang/LLVM 12.0.0. SPECint[®]_rate_base_2017 compiler switches: Intel(R) oneAPI DPC++/C++ Compiler: -xCORE-AVX512 -O3 -ffast-math -flto -mfpmath=sse -funroll-loops -qopt-mem-layout-trans=4. Intel(R) C++ Intel(R) 64 Compiler Classic: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-mem-layout-trans=4 -qopt-multiple-gather-scatter-by-shuffles. GCC: -march=skylake-avx512 -mfpmath=sse -Ofast -funroll-loops -flto -mprefer-vector-width=128. LLVM: -march=skylake-avx512 -mfpmath=sse -Ofast -funroll-loops -flto. qkmallo used for intel compiler. jemalloc 5.0.1 used for gcc and llvm. SPECfp[®]_rate_base_2017 compiler switches: Intel(R) oneAPI DPC++/C++ Compiler: -xCORE-AVX512 -Ofast -ffast-math -flto -mfpmath=sse -funroll-loops -qopt-mem-layout-trans=4. Intel(R) C++ Intel(R) 64 Compiler Classic: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-prefetch -ffinite-math-only -qopt-multiple-gather-scatter-by-shuffles -qopt-mem-layout-trans=4. GCC: -march=skylake-avx512 -mfpmath=sse -Ofast -fno-associative-math -funroll-loops -flto. LLVM: -march=skylake-avx512 -mfpmath=sse -Ofast -funroll-loops -flto.

SPECspeed* 2017 (推定値)

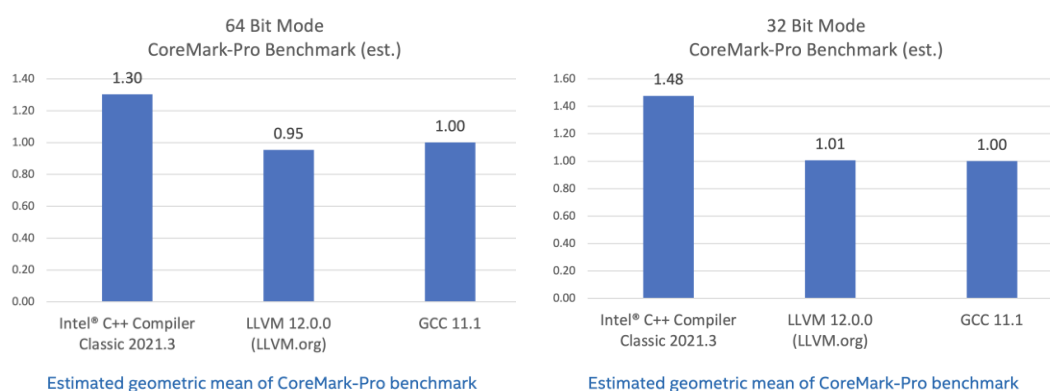


SPEC CPU* 2017 ベンチマーク・パッケージには、システムのプロセッサ、メモリー・サブシステム、コンパイラに負荷をかけ、計算集約型のパフォーマンスを測定して比較する、業界標準の CPU 集約型のスイートが含まれています。SPEC ベンチマークの詳細は、<https://www.spec.org> (英語) を参照してください。

Configuration: Testing by Intel as of Jun 10, 2021. Intel(R) Xeon(R) Platinum 8380 CPU @ 2.30GHz, 2 socket, Hyper Thread on, Turbo on, 32G x16 DDR4 3200 (1DPC). Red Hat Enterprise Linux release 8.2 (Ootpa), 4.18.0-193.el8.x86_64. Software: Intel(R) oneAPI DPC++/C++ Compiler for applications running on Intel(R) 64, Version 2021.3.0 Build 20210604. Intel(R) C++ Intel(R) 64 Compiler Classic for applications running on Intel(R) 64, Version 2021.3.0 Build 20210604_000000, GCC 11.1, Clang/LLVM 12.0.0. SPECint[®]_speed_base_2017 compiler switches: Intel(R) oneAPI DPC++/C++ Compiler: -xCORE-AVX512 -O3 -ffast-math -flto -mfpmath=sse -funroll-loops -qopt-mem-layout-

trans=4 -fopenmp. Intel(R) C++ Intel(R) 64 Compiler Classic: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-mem-layout-trans=4 -qopt-GCC: -march=skylake-avx512 -mfpmath=sse -Ofast -funroll-loops -flto -fopenmp. LLVM: -march=skylake-avx512 -mfpmath=sse -Ofast -funroll-loops -flto -fopenmp=libomp. multiple-gather-scatter-by-shuffles -qopenmp. jemalloc 5.0.1 used for intel compiler, gcc and llvm. SPECfp®_speed_base_2017 compiler switches: Intel(R) oneAPI DPC++/C++ Compiler: -xCORE-AVX512 -Ofast -ffast-math -flto -mfpmath=sse -funroll-loops -qopt-mem-layout-trans=4 -fopenmp. Intel(R) C++ Intel(R) 64 Compiler Classic: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-prefetch -ffinite-math-only -qopt-multiple-gather-scatter-by-shuffles -qopenmp. GCC: -march=skylake-avx512 -mfpmath=sse -Ofast -fno-associative-math -funroll-loops -flto -fopenmp. LLVM: -march=skylake-avx512 -mfpmath=sse -Ofast -funroll-loops -flto -fopenmp=libomp. jemalloc 5.0.1 used for intel compiler, gcc and llvm.

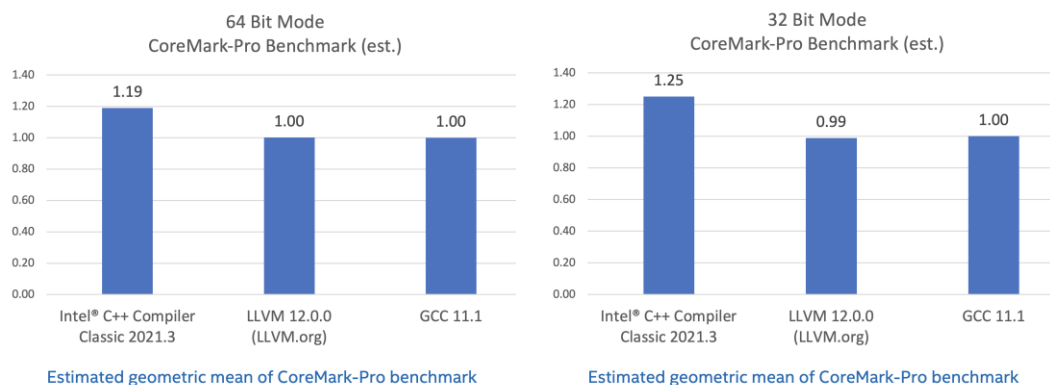
インテル® Core™ i7-8700K プロセッサ上での CoreMark*-Pro



CoreMark*-Pro は、マルチコア・テクノロジー、整数型と浮動小数点型のワークロードの組み合わせ、大規模メモリー・サブシステムを利用するデータセットなどをサポートしており、プロセッサ全体をテストすることを目的としています。Embedded Microprocessor Benchmark Consortium (EEMBC) の CoreMark*-Pro の詳細は、<https://www.eembc.org/coremark-pro/> (英語) を参照してください。

Testing by Intel as of Jun 10, 2021 - Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz, 16G x2 DDR4 2666. Software: Intel(R) C++ Compiler Classic for applications running on Intel(R) 64, Version 2021.3.0 Build 20210604_000000, GCC 11.1, Clang/LLVM 12.0.0. Red Hat Enterprise Linux release 8.0 (Ootpa), 4.18.0-80.el8.x86_64. Compiler switches: Intel(R) C++ Compiler Classic for applications running on Intel(R) 64, Version 2021.1 Build 20201112_000000: icc -xCORE-AVX2 -mtune=skylake -ipo -O3 -no-prec-div -qopt-prefetch. GCC 11.1: gcc -march=native -mfpmath=sse -Ofast -funroll-loops -flto. LLVM 12.0.0: clang -Ofast -funroll-loops -flto -static -mfpmath=sse -march=native.

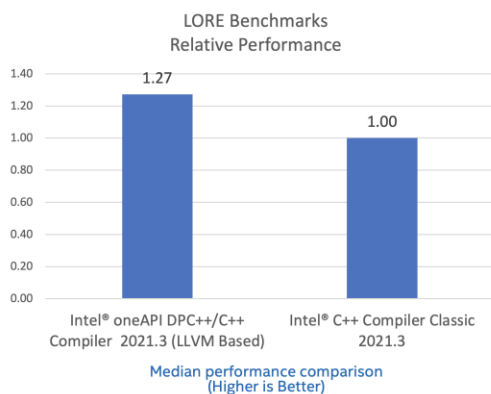
Intel Atom® C3850 プロセッサ上での CoreMark*-Pro



CoreMark*-Pro は、マルチコア・テクノロジー、整数型と浮動小数点型のワークロードの組み合わせ、大規模メモリ・サブシステムを利用するデータセットなどをサポートしており、プロセッサ全体をテストすることを目的としています。Embedded Microprocessor Benchmark Consortium (EEMBC) の CoreMark*-Pro の詳細は、<https://www.eembc.org/coremark-pro/> (英語) を参照してください。

Configuration: Testing by Intel as of Jun 10, 2021 - Intel(R) Atom(TM) CPU C3850 @ 2.10GHz, 16G x2 DDR4 2400. Software: Intel(R) C Intel(R) 64 Compiler Classic for applications running on Intel(R) 64, Version 2021.1 Build 20201112_000000, GCC 11.1, Clang/LLVM 12.0.0. Red Hat Enterprise Linux release 8.0 (Ootpa), 4.18.0-80.el8.x86_64. Compiler switches: Intel(R) C++ Compiler Classic for applications running on Intel(R) 64, Version 2021.1 Build 20201112_000000: `icc -xATOM_SSE4.2 -mtune=goldmont -ipo -O3 -no-prec-div -qopt-prefetch`. GCC 11.1: `gcc -march=native -mfpmath=sse -Ofast -funroll-loops -flto`. LLVM 12.0.0: `clang -Ofast -funroll-loops -flto -static -mfpmath=sse -march=native`.

LORE (Loop Repository for Evaluation of Compilers) ベンチマーク

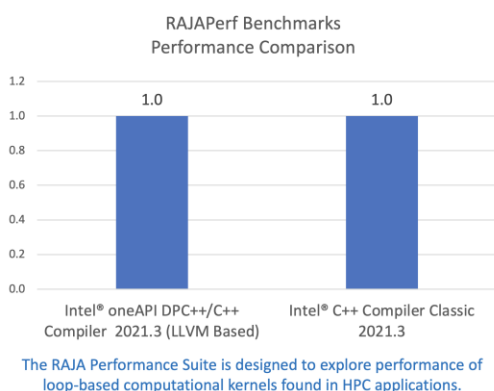


LORE は、一般的なベンチマークやライブラリー、実際のアプリケーションから抽出したループネストに対して、C 言語のテストを行います。ループは、コンパイラー・コミュニティがループの最適化を評価するために使用できるさまざまな特性をカバーしています。ここでは、65 のベンチマークとワークロードをテストしました。詳細は、<https://www.vectorization.computer> (英語) を参照してください。

Configuration: Testing by Intel as of Jun 9, 2021 - Intel(R) Xeon(R) Platinum 8180CPU @ 2.50GHz, 2 socket, 28 cores, HT enabled, Turbo enabled, 384GB RAM. Software: Intel(R) oneAPI DPC++/C++

Compiler for applications running on Intel(R) 64, Version 2021.2.0 Build 20210607, Intel(R) C++ Compiler Classic for applications running on Intel(R) 64, Version 2021.3.0 Build 20210607. Ubuntu 18.04.1 with GCC 10.2.0. Compiler switches: Intel(R) C++ Compiler Classic for applications running on Intel(R) 64, Version 2021.3.0 Build 20210607: ICC OPT - OPT="-Ofast -qopt-prefetch -unroll-aggressive -restrict -xHost -w". ICC OPT512 - OPT="-Ofast -qopt-prefetch -unroll-aggressive -restrict -xHost -w -qopt-zmm-usage=high". Intel(R) oneAPI DPC++/C++ Compiler for applications running on Intel(R) 64, Version 2021.2.0 Build 20210607: ICX OPT - OPT="-Ofast -qopt-prefetch -unroll-aggressive -restrict -xHost -w". ICX OPT512 - OPT="-Ofast -qopt-prefetch -unroll-aggressive -restrict -xHost -w -mprefer-vector-width=512". ICX OPTm - OPT="-Ofast -qopt-prefetch -unroll-aggressive -restrict -march=skylake-avx512 -w". ICX OPT512m - OPT="-Ofast -qopt-prefetch -unroll-aggressive -restrict -march=skylake-avx512 -w -mprefer-vector-width=512.

RAJA Performance Suite (RAJAPerf)



RAJA Performance Suite は、HPC アプリケーションにおけるループベースの計算カーネルのパフォーマンスを調査します。RAJA Performance Suite の詳細は、<https://github.com/LLNL/RAJAPerf> (英語) を参照してください。

このベンチマークでは、インテル® C/C++ コンパイラー・クラシックと同等のパフォーマンスが測定されました。上回ってはいないものの、素晴らしい結果であると言えます。これらのベンチマークは、LLVM ベースの新しいバージョンのインテル® C/C++ コンパイラーが全面的に推奨できること確認するためのものであるため、ここではこのベンチマークもあえて掲載しました。

Configuration: Testing by Intel as of Jun 9, 2021 -Intel(R) Xeon(R) Platinum 8180CPU @ 2.50GHz, 2 socket, 28 cores, HT enabled, Turbo enabled, 384GB RAM. Software: Intel(R) oneAPI DPC++/C++ Compiler for applications running on Intel(R) 64, Version 2021.2.0 Build 20210607, Intel(R) C++ Compiler Classic for applications running on Intel(R) 64, Version 2021.3.0 Build 20210607. Ubuntu 18.04.1 with GCC 10.2.0. Compiler switches: Intel(R) C++ Compiler Classic for applications running on Intel(R) 64, Version 2021.3.0 Build 20210607: ICC OPT OPT="-Ofast -ansi-alias -xCORE-AVX512", ICC OPT512 OPT="-Ofast -ansi-alias -xCORE-AVX512 -qopt-zmm-usage=high", setenv KMP_AFFINITY compact,granularity=fine. Intel(R) oneAPI DPC++/C++ Compiler for applications running on Intel(R) 64, Version 2021.2.0 Build 20210607: ICX OPT OPT="-Ofast -ansi-alias -xCORE-AVX512", ICX OPT512 OPT="-Ofast -ansi-alias -xCORE-AVX512 -qopt-zmm-usage=high", setenv KMP_AFFINITY compact,granularity=fine

性能は、使用状況、構成、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。性能の測定結果はシステム構成の日付時点のテストに基づいています。また、現在公開中のすべてのセキュリティ・アップデートが適用されているとは限りません。詳細は、システム構成を参照してください。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。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。

製品および性能に関する情報

¹ 性能は、使用状況、構成、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。