

NVIDIA* GPU と AMD* GPU の SYCL* パフォーマンスはネイティブシステム言語に匹敵

この記事は 2023 年 4 月 6 日に Codeplay のウェブサイトで公開された「[SYCL™ Performance for Nvidia® and AMD GPUs Matches Native System Language](#)」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。

NVIDIA* GPU と AMD* GPU 上でワークロードを実行するベンチマークにおいて、oneAPI の SYCL* 実装である DPC++ は、ネイティブの CUDA* と HIP に匹敵するパフォーマンスを達成しました。

Codeplay は長年にわたって、[Khronos Group](#) のオープン・スタンダードである [SYCL* プログラミング・モデル](#) (英語) を主導し貢献するなど、ヘテロジニアス・プログラミングの発展に取り組んできました。当初から私たちは、開発者の労力を最小限に抑えながら、いかにしてマルチベンダーのヘテロジニアス・ハードウェアに対応するかに注目してきました。マルチベンダーの世界で生産性とパフォーマンスを実現するのは容易ではありません。すべてのアクセラレーター・ベンダーは、ハードウェア機能により他社との差別化を図り、通常はネイティブの (そして独自の) プログラミング・モデルを使ってこれらの機能を公開しています。SYCL* を設計する際、私たちは、市場のすべてのベンダーのさまざまなハードウェア機能を簡単に利用できる業界標準とすることに力を注ぎました。これにより、SYCL* アプリケーションは、複数ベンダーのアーキテクチャーにわたって真にポータブルであると同時に、カスタム・アーキテクチャーのチューニングも可能にし、長期にわたって安定したソリューションを提供します。

懐疑的になるのは無理ありません。ネイティブ・システム・プログラミングが確立されており、単一アーキテクチャー・システムで問題なく機能している場合は特にそうです。私自身も懐疑的でした。[oneAPI](#) (英語) と SYCL* を使うことで、さまざまなタイプのワークロードで明らかなメリットが得られます。より多くの企業、組織、スーパーコンピューティング・センターが、[インテル®](#) アーキテクチャーだけでなく、NVIDIA* GPU や AMD* GPU を含むほかのアーキテクチャーでも oneAPI を採用しています。複数のベンダーのアーキテクチャーで単一のコードベースを使用できるようになれば、移植性と生産性において大きなメリットが得られることが分かっています。まだ始まったばかりですが、NVIDIA* GPU や AMD* GPU 上で oneAPI¹ によって最適化された SYCL* ワークロードが、ネイティブシステム言語 (NVIDIA の場合は CUDA*、AMD の場合は HIP) と比較して、同等かそれ以上のパフォーマンスを示す実際のベンチマークや数値が示されています。

SYCL*: 優れたパフォーマンスと移植性を実現

SYCL* は、[インテル](#)、AMD、NVIDIA、および業界のその他のベンダーの複数のアクセラレーター (CPU、GPU、FPGA) 上で実行される C++ ベースの並列プログラミング言語です。複数の計算アーキテクチャー間で移植可能であることが証明されており、それらの計算ユニット上のネイティブおよび確立されたプログラミング環境と同等のパフォーマンスを提供します。SYCL* は「魔法」ではなく、CUDA*、oneTBB、その他の並列プログラミング・モデルと同様の概念を備えた実証済みのテクノロジーです。開発者は、メモリー割り当て、キューの送信、コードのパフォーマンスに重要な部分のカーネルなど、標準的な概念を使用してコードを記述します。DPC++ のような SYCL* 実装は、すべての要素を 1 つのアプリケーション上にまとめて、単一のコンパイラー呼び出しから複数のシステム上で実行できるようにします。これはすべて、独自の構文を使用しない標準 C++ コードのみを使用して行われます。

パフォーマンス・ベンチマーク: SYCL* と CUDA* および HIP

XPU アクセラレーションにおいてパフォーマンスは非常に重要な要素であり、oneAPI がターゲットとするハードウェアから現時点で可能な限りのパフォーマンスを引き出せるよう、私たちは多大な労力を費やしてきました。いくつかのベンチマーク結果は、多くのケースでネイティブ・パフォーマンスに匹敵することを示しており、今後もこの取り組みは続けていきます。私たちの経験から、大半のユースケースでは、プログラミング・モデルの基本的な部分でパフォーマンスに差が生じることはなく、パフォーマンス問題の大半は、異なる値のパフォーマンス・チューニング (例えば、CUDA* ブロックサイズと SYCL* のローカル・ワークグループ) や、インライン展開やアンロールなどコード生成に影響を与えるツールチェーン・オプションによるもので、これらは簡単にチューニングできます。SYCL* のおかげで、複数のターゲットで優れたパフォーマンスを達成できる単一のソースを維持することが可能です。特定のベンダーのハードウェア機能は、拡張機能により使用できます。また、SYCL* は、コードを中立に保ちながら、既存のベンダー最適化ライブラリーと相互運用できます。前述のとおり、魔法ではなく、健全なエコシステムを構築するオープン・スタンダードと強力なコミュニティの関与だけです。

図 1 (下) は、オープンソースの [SYCLomatic ツール](#) (英語) を使用して CUDA* コードを SYCL* に移行し、DPC++/C++ CUDA* バックエンドを使用して同じ NVIDIA* GPU 上でコンパイルして実行した場合のパフォーマンス比較を示しています。これは、移行がほぼ透過的であり、生成される SYCL* コードが元の CUDA* コードと同等のパフォーマンスを発揮するコード移行の良い例です。ソースコードは、スクリプトとともに [GitHub* リポジトリ](#) (英語) にあるので、このプロジェクトを自分で試して実験することができます。



図 1. SYCLomatic を使用して CUDA* から SYCL* に移行した NBODY シミュレーションの出力。CUDA* バージョンはカーネル内で平均 6.7ms を費やしているのに対し、CUDA* バックエンドで動作する SYCL* バージョンは平均 6.5ms を費やしています。†

†システム構成: Lenovo* ThinkBook 16p RTX3060 Laptop (AMD Ryzen* 7 5800H Processors, NVIDIA* GeForce RTX* 3060) with Ubuntu* 20.04, DPC++ built from source and CUDA* 16)。使用したコンパイラーオプションは [GitHub* プロジェクトのビルドスクリプト](#) (<https://github.com/codeplaysoftware/cuda-to-sycl-nbody>) にあります。

図 2 は、さまざまなアプリケーション・ドメインのさまざまなベンチマーク含むオープンソースの [HeCBench プロジェクト](#) (英語) から代表的なベンチマークを抜粋し、DPC++ とネイティブ CUDA* のパフォーマンスを示しています。これらのベンチマークのほとんどは、SYCL* とネイティブ CUDA* のパフォーマンスが同等であるこ

とを示しています。一部のケースでは、SYCL* コードのほうが高速または低速であり、さらにパフォーマンスを調整することができます。

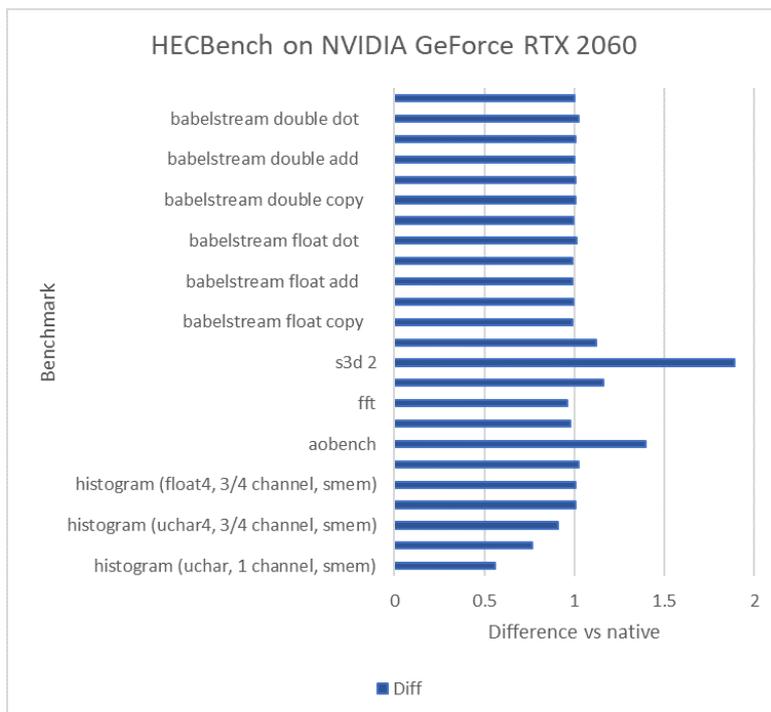
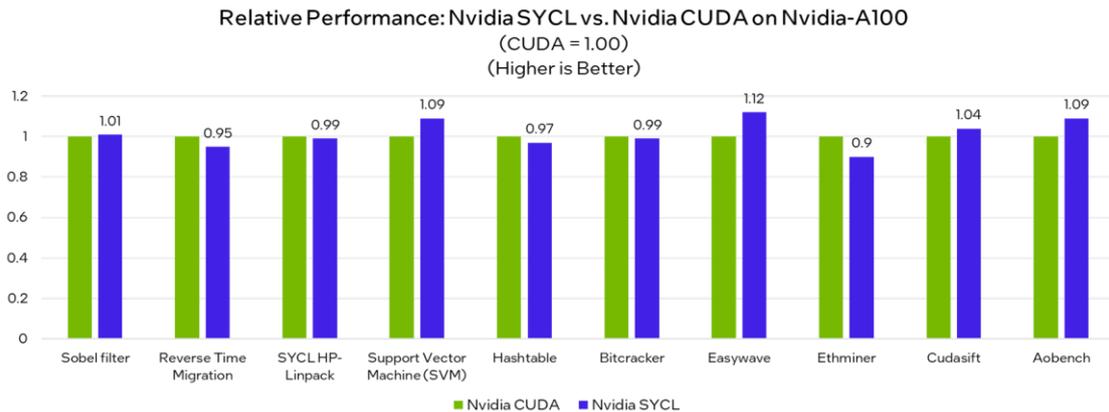


図 2. 2022 年 12 月時点の測定値。NVIDIA* GeForce RTX* 2060、CUDA* 11.7 上でインテル® oneAPI ベース・ツールキット 2023.0、NVIDIA* GPU 向け oneAPI 2023.0 により最適化された HECBench を実行したネイティブ CUDA* と CUDA* 上での SYCL* のパフォーマンス比較。

github.com/zjin-lcf/HeCBench (英語)

次に、SYCL* で記述された多くの業界ドメインにまたがるアルゴリズム/データセットと、ネイティブシステム言語である CUDA* (NVIDIA* GPU) および HIP (AMD* GPU) の実装を比較したパフォーマンス・ベンチマークを示します。各データセットは、計算アーキテクチャーとネイティブ言語ごとに特性評価、分析、およびチューニングが行われました。図 3 は、NVIDIA* A100 システム上で SYCL* と CUDA* のパフォーマンスを比較した 10 個のワークロードを示しています。6 つのワークロードでは SYCL* のパフォーマンスが CUDA* を上回り、残りのワークロードではパフォーマンスの差はわずかです。

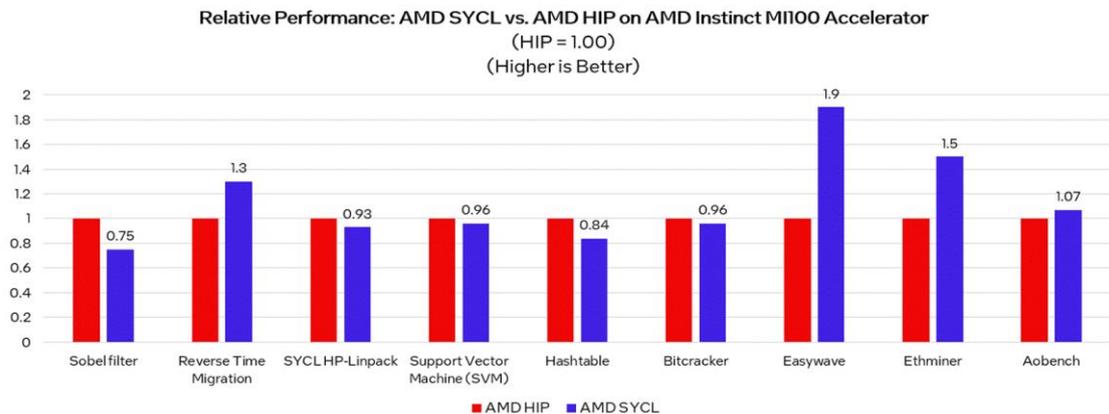


Testing Date: Performance results are based on testing by Intel as of August 15, 2022 and may not reflect all publicly available updates.
Configuration Details and Workload Setup: Intel® Xeon® Platinum 8360Y CPU @ 2.4GHz, 2 socket, Hyper Thread On, Turbo On, 256GB Hynix DDR4-3200, ucode 0x000363, GPU: Nvidia A100 PCIe 80GB GPU memory. Software: SYCL open source/CLANG 15.0.0, CUDA SDK 11.7 with NVIDIA NVCC 11.7.64, cuMath 11.7, cuDNN 11.7, Ubuntu 22.04.1. SYCL open source/CLANG compiler switches: -fsycl-targets=nvptx64-nvidia-cuda, NVIDIA NVCC compiler switches: -O3 -gencode arch=compute_80,code=sm_80. Represented workloads with Intel optimizations.
 Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See configuration disclosure for details. No product or component can be absolutely secure.
 Performance varies by use, configuration, and other factors. Learn more at www.intel.com/PerformanceIndex. Your costs and results may vary.

図 3. NVIDIA® A100 上で SYCL® と CUDA® で実行した一部のデータセットの相対パフォーマンス比較。
 6 つのワークロードでは、SYCL® のパフォーマンスは CUDA® を上回るか、同等です。
 その他のワークロードのパフォーマンスの差はわずかです²。

図 4 は、AMD Instinct® MI100 システム上では、9 つのワークロードで SYCL® のパフォーマンスが HIP に匹敵することを示しています。SYCL® と CUDA® および HIP を比較したベンチマーク・テストの結果は、コンパイラおよびランタイム・ツールチェーンの成熟度と機能の違いに起因しています。

Relative Performance: AMD SYCL vs. AMD HIP on AMD GPU



Testing Date: Performance results are based on testing by Intel as of August 15, 2022 and may not reflect all publicly available updates.
Configuration Details and Workload Setup: Intel® Xeon® Gold 6330 CPU @ 2.0GHz, 2 socket, Hyper Thread Off, Turbo On, 256GB Hynix DDR4-3200, ucode 0x000363, GPU: AMD Instinct MI100, 32GB GPU memory. Software: SYCL open source/CLANG 15.0.0, AMD ROCm 5.2.1 with AMD-HIPCC 5.2.21152-4b155a06, hipSolver 5.2.1, rocBLAS 5.2.1, Ubuntu 20.04.4. SYCL open source/CLANG compiler switches: -fsycl-targets=amdgc-nv-amd-amdhsa-Xsycl-target-backend-offload-arch=gfx908, AMD-HIPCC compiler switches: -O3. Represented workloads with Intel optimizations.
 Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See configuration disclosure for details. No product or component can be absolutely secure.
 Performance varies by use, configuration, and other factors. Learn more at www.intel.com/PerformanceIndex. Your costs and results may vary.

図 4. AMD Instinct® MI100 アクセラレーター上で SYCL® と HIP で実行した 9 つのデータセットの相対パフォーマンス比較³。

マルチアーキテクチャーのスーパーコンピューティングが道を開く

oneAPI は、マルチアーキテクチャーのスーパーコンピューターがエクサスケールやゼタスケールに移行するのを支援しています。アルゴンヌ国立研究所は、[インテル® データセンター GPU マックス・シリーズを使用した初期のパフォーマンス値 \(英語\)](#) を発表し、[ケンブリッジ大学 \(英語\)](#) と [テキサス先端コンピューティング・センターの Frontera スーパーコンピューター \(英語\)](#) は、oneAPI がサポートするいくつかの大きなプロジェクトを主導しています。

米国 Exascale Computing Project (ECP) の 2023 Community Birds-of-a-Feather (BOF) Days の [HPCWire の記事「SYCL* Progress and Performance \(SYCL* の進捗とパフォーマンス\)」 \(英語\)](#) では、Aurora に関する Argonne Leadership Computing Facility (ALCF)、米国初のエクサスケール・システム Frontier に関する Oak Ridge Leadership Computing Facility (OLCF)、Perlmutter に関する National Energy Research Scientific Computing Center (NERSC) などのプログラミングに関する情報を共有されました。

時間を測定するワークロード (デバイス上で費やされる時間とホストとデバイス間のデータ転送にかかる時間として定義される) については、HPC、ビッグデータ計算、マシンラーニングとディープラーニング、レイトレーシング、暗号/ビットマイニングなどのドメインが選択されました。各測定は 10 回実行され、最初の測定は破棄して、残りで平均が計算されました。ほとんどの場合、パフォーマンスの向上は主にコンパイラーによる効率良いコード生成と、セットアップ/初期化の時間差に起因していました。

まとめ

これらの結果から、SYCL* は NVIDIA* と AMD* のデバイス上で高いパフォーマンスを発揮し、さまざまなワークロードでネイティブの CUDA* や HIP コードのパフォーマンスに匹敵することが分かります。SYCL* と oneAPI の開発環境、コンパイラー、ツール、ライブラリーは、非常に効率良く競争力があります。SYCLomatic のようなコード移行ツールは、CUDA* から SYCL* へのコードの移行を容易にします。

マルチアーキテクチャー、マルチベンダーのプログラミング、oneAPI と SYCL* の利用が拡大するにつれ、主要なスーパーコンピューターや移行されたアプリケーションによる実際の使用が道を切り開いています (詳細は後述)。今後、さらに多くのパフォーマンス比較が行われることでしょう。そのような比較結果を目にし、エコシステムと協力して、すべての人のためのオープンな標準ベースのヘテロジニアス・コンピューティングを推進することを楽しみにしています。

NVIDIA* GPU および AMD* GPU 上で SYCL* を使用してみたい方は、[Codeplay の開発者向けウェブサイト \(英語\)](#) から NVIDIA* GPU および AMD* GPU 向け oneAPI プラグインをダウンロードしてください。

関連情報

- [oneAPI \(英語\)](#)
- [NVIDIA* GPU および AMD GPU 向け oneAPI プラグイン \(英語\)](#)
- [CUDA* から SYCL* への移行ツール: オープンソースの SYCLomatic プロジェクト \(英語\) | インテル® DPC++ 互換性ツール \(インテル® DPCT\) \(英語\)](#)
- iSUS での日本語情報:
 - [NVIDIA* GPU 向け oneAPI プラグイン](#)
 - [AMD GPU 向け oneAPI プラグイン](#)

- [インテル® DPC++ 互換性ツール 2023.2 デベロッパー・ガイドおよびリファレンス](#)

法務上の注意書き

¹ oneAPI の SYCL* 実装はデータ並列 C++ (DPC++) です。

² 図 3. 相対パフォーマンス: NVIDIA* A100 上の NVIDIA* SYCL* と NVIDIA* CUDA* – 性能の測定結果は 2022 年 8 月 15 日現在のインテルの社内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。システム構成とワークロード設定: Intel® Xeon® Platinum 8360Y CPU @ 2.4GHz, 2 socket, Hyper Thread On, Turbo On, 256GB Hynix DDR4-3200, ucode 0x000363。GPU: NVIDIA* A100 PCIe* 80GB GPU memory。ソフトウェア: SYCL* open source/CLANG 15.0.0, CUDA* SDK 11.7 with NVIDIA-NVCC 11.7.64 cuMath 11.7, cuDNN 11.7, Ubuntu* 22.04.1。SYCL* オープンソース/CLANG コンパイラー・オプション: -fsycl-targets=nvptx64-nvidia-cuda, NVIDIA NVCC compiler switches: -O3-gencode arch=compute_80, code=sm_80。インテルにより最適化された代表的なワークロード。

³ 図 4. 相対パフォーマンス: AMD Instinct* MI100 アクセラレーター上の AMD* SYCL* と AMD* HIP – 性能の測定結果は 2022 年 8 月 15 日現在のインテルの社内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。システム構成とワークロード設定: Intel® Xeon® Gold 6330 CPU @ 2.0GHz, 2 socket, Hyper Thread Off, Turbo On, 256GB Hynix DDR4-3200, ucode 0xd000363。GPU: AMD Instinct* MI100, 32GB GPU memory。ソフトウェア: SYCL* open source/CLANG 15.0.0, AMD* RoCm* 5.2.1 with AMD-HIPCC 5.2.21152-4b155a06, hipSolver 5.2.1, rocBLAS 5.2.1, Ubuntu* 20.04.4。SYCL* open source/CLANG compiler switches: -fsycl-targets=amdgc-nv-amdhsa-Xsycl-target-backend-offload-arch=gfx908, AMD-HIPCC compiler switches: -O3。インテルにより最適化された代表的なワークロード。

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。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、<http://www.intel.com/PerformanceIndex/> (英語) を参照してください。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。

Codeplay Software Ltd has published this article only as an opinion piece. Although every effort has been made to ensure the information contained in this post is accurate and reliable, Codeplay cannot and does not guarantee the accuracy, validity or completeness of this information. The information contained within this blog is provided "as is" without any representations or warranties, expressed or implied. Codeplay Software Ltd makes no representations or warranties in relation to the information in this post.