

ヘテロジニアスな処理にはデータ並列化が必要: SYCL と DPC++ から始めよう

この記事は、Intel Tech.Decoded で公開されている「[How to Build Applications with Intel® Software to Get Best Performance on Intel® Hardware](#)」の日本語参考訳です。

[注: この記事は、[The New Stack](#) (英語) で掲載されている James Reinders の記事を転載したものです。]

これからは XPU の時代です。

ハードウェアの革新により、CPU、GPU、FPGA、DSP、ASIC などが爆発的に増えていますが、これらは XPU と総称することができます。XPU とは、あらゆる種類の「処理ユニット」であり、アプリケーションの計算を支援するあらゆるハードウェアを指します。

XPU の急増は、より多様な処理ユニットに対応したコードが求められるため、開発者にとって大きな課題です。新しいアーキテクチャーでアプリケーションのパフォーマンスを向上するには、コードを書き換えたり、テストしたりする時間やコストを考慮しなければなりません。コードの保守性を維持するため、コードをできるだけ多くの XPU に対応させることが最も重要です。アプリケーション開発をクロスアーキテクチャー・モデルに移行することで、時間とコストを大幅に削減できることが明らかになっていますが、ヘテロジニアス・コンピューティングの普及に伴い、これは差し迫った課題となっています。

現在、私たちの世界は XPU の世界へと急速に変化しており、最終的にはすべてのコンピューティングを変革することになるため、再考が必要です。

XPU: 高速な計算を行うソフトウェアの再発明

広く利用されているソフトウェア・プログラミング・システムである CUDA* は、NVIDIA* GPU 向けに設計されており、効果を発揮しています。OpenCL* はオープンなアプローチで、マルチベンダー・サポートを実現していますが、C 言語中心で C++ のニーズに対応できないという欠点があります。

CUDA* と OpenCL* は、その目的を十分に果たしてきましたが、今後は XPU の可能性を引き出すため、真にオープンでマルチベンダーなアプローチが必要となります。

SYCL とデータ並列 C++ (DPC++) が最善である理由

CUDA* と OpenCL* から得た経験を基に、ヘテロジニアス・システム向けの C++ ベースのデータ並列化を実現する、真にオープンなソリューションが登場しました。それが SYCL です。SYCL は、複数のハードウェア・アクセラレーター上でプログラミングの生産性を向上させる、高度なプログラミング・モデルです。SYCL* は、さま

さまざまなベンダーによるサポート、幅広い関心、そして複数のコンパイラー・プロジェクトによるサポートを短期間で達成しました。

ヘテロジニアス化が進む世界で効率良くプログラミングを行うには、すべての XPU でパフォーマンスを引き出す必要があるため、SYCL* は重要です。真にオープンなアプローチだけがそれを可能にします。

SYCL は、ヘテロジニアスなハードウェア(XPU)を対象としたシングルソースの C++ データ並列プログラミングを実現するオープン・スタンダードです。SYCL は、ホストに C++ を使用し、デバイスにドメイン固有のカーネル言語を使用するのではなく、システム上の複数のデバイスを対象に C++ でシングルソース・コンパイルを可能にします。

SYCL は、C++ にカーネルスタイルのプログラミングと、システム内のアクセラレーターを検索、照会、使用するためのメカニズムをもたらします。カーネルベースのプログラミングは、データ並列性を活用する重要なプログラミング・スタイルであり、OpenCL* や CUDA*でもサポートされています。標準的な方法でアクセラレーターを列挙してアクセスする機能は、OpenCL*でも導入されています。

また、DPC++ は、LLVM コミュニティーにオープンな実装を提供しており、すべてを LLVM C++コンパイラーに反映することを目指しています。DPC++ は、いくつかの拡張を加えて SYCL を実装することを目的としており、[SYCL 2020](#) (英語) 標準規格がリリースされる前から、その多くの機能の実装に取り組んできました。現在、SYCL 2020 仕様全体との整合性をとるための作業を行っており、すべての作業は非常に活発な [オープンソース・リポジトリ](#) (英語) で確認できます。DPC++ は、[インテル](#) (英語) がインテルの CPU、GPU、FPGA ターゲットに使用しています。[Codeplay*](#) (英語) も NVIDIA* GPU ターゲットに使用しています。別の SYCL コンパイラーである [hipSYCL](#) (英語) は、AMD の HIP/ROCm* と接続することで、AMD* の CPU や GPU をサポートしています。SYCL 用の複数のオープンソース・コンパイラーが存在することは、コミュニティにとって素晴らしいことであり、SYCL が幅広く、多様で、オープンなサポートを受けていることを示しています。

2019 年から 2020 年にかけて、私はチームと協力して、SYCL と DPC++ に関する初めての書籍を出版しました。この書籍は、[Apress のウェブサイト](#) (英語) から無料でダウンロードできます。書籍の出版後まもなく、Khronos Group* が [SYCL 2020](#) (英語) の最終仕様を発表しました。

SYCL 2020 の仕様がリリースされたことは、重要なマイルストーンと言えます。この仕様は、業界の多くの熱心な人々による長年の仕様策定の成果であり、将来性のあるオープンな仕様です。C++17 をベースにした SYCL 2020 は、標準的な C++ アプリケーションの高速化を可能にし、ISO C++ のロードマップとの連携を強化しています。Khronos Group* は SYCL 2020 の発表の中で、統合共有メモリー (USM) のサポート、ビルトイン・リダクション、CTAD の広範な使用、C++ 標準に沿ったアトミック操作など、SYCL 2020 の特徴を強調しています。

多様性とプログラミングの健全性のため XPU のオープン性を維持しよう

SYCL と DPC++ は、XPU を効率良く利用する手助けとなります。これらは、ライブラリーやさまざまなソフトウェア開発ツールを含む、XPU をサポートするための広範な推進活動の一部です。これこそが、私が情熱を持って取り組んでいる [oneAPI 業界イニシアチブ](#) (英語) の原点であり、インテルに復帰した際には、その一員

となることを楽しみにしていました。すべての XPU をオープンに使用する課題を軽減する、という目的への支援が SYCL と oneAPI への関心を高めています。その一例が、oneAPI ディープ・ニューラル・ネットワーク・ライブラリー (oneDNN) です。当初、oneDNN はインテル・プロセッサ向けに高度に最適化されていましたが、世界最速のコンピューターである日本の「富岳」(ARM* プロセッサ搭載) を加速させました。その結果、oneDNN は現在、ARM* も強力にサポートしています。オープンな SYCL と oneAPI のライブラリーとツールは、オープン性とパフォーマンスを備えた、すべての XPU 向けのプログラミングを実現する新しい時代の到来を後押ししています。

ソフトウェア開発者のコミュニティが一丸となって、業界全体に貢献する SYCL をはじめとする標準規格を策定し、ヘテロジニアス・プログラミング (XPU) や、並列処理を取り入れた最新の C++ の採用を強力にサポートすることができます。

SYCL は、幅広いサポート、多くの参加者、複数のオープンソースの実装、そして無限の可能性を秘めたオープン・スタンダードです。DPC++ は、オープンな LLVM ベースのコンパイラーを提供し、SYCL をサポートする労力を軽減し、XPU 間の強力な互換性を促進します。oneAPI は、ソフトウェア開発のあらゆる側面において、XPU のオープンでハイパフォーマンスなアクセスについて議論し、推進するためのフォーラムを提供しています。

XPU 時代に備え、SYCL、DPC++、oneAPI について理解を深めていただければと思います。私たちは一丸となってオープンに XPU をサポートし、XPU の多様性から得られるメリットを享受し、効率良くプログラミングを行いましょ。

Intel、インテル、Intel ロゴ、その他のインテルの名称やロゴは、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。

© 2021 Intel Corporation.