多様なハードウェアにおける HPC コードの移植性

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「Expand HPC Code Portability Across Diverse Hardware Architectures」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。

課題

今日の多様なコンピューター・アーキテクチャーの中で、効率的でスケーラブル、かつ選択的なターゲットに最適化されたコードを開発することは、開発者にとって継続的な課題です。最近、ハイパフォーマンス・コンピューティング (HPC) が脚光を浴びています。HPC は、AI、ビジュアル・コンピューティング、データ・アナリティクス、ビデオ分析などのテクノロジーにまたがる要求の厳しいワークロードをサポートします。また、ハイブリッド・クラウドやエッジ・コンピューティングなどの新しいテクノロジーもサポートします。

最高のパフォーマンス・レベルを維持しながら複数のアーキテクチャー・モデルの個々の特性に合わせたコードを開発するには、CPU、GPU、AI、FPGA などのアクセラレーターに実装されたスカラー、ベクトル、行列、空間 (SVMS) アーキテクチャーの組み合わせに対応する必要があります。以前は、この目的を達成するため、多くの異なる言語、ライブラリー、およびツールが必要でした。また、ターゲットとなるプラットフォームごとに、個別のプログラミング・パスを構築しなければならないこともしばしばでした。異なるアーキテクチャーでパフォーマンスを最適化するには別々のソフトウェアに投資する必要があり、コードの移植性と再利用性の妨げとなっていました。

シュトゥットガルト大学の博士課程に在籍する Marcel Breyer 氏は、この課題に興味を持ち、解決策を見出すことに意欲を燃やしていました。そして、C++、OpenMP*、メッセージ・パッシング・インターフェイス (MPI)、CUDA*、SYCL* など、HPC のユースケースに最適なツールとテクノロジーを特定して適用する取り組みを始めました。彼の研究は、コードの移植性についてより深い洞察をもたらしました。また、クロスプラットフォームの相互運用性を確保し、ヘテロジニアス・システムをプログラミングする際の多くの困難を解決するのに、oneAPI プログラミング・モデルが有効であることが確認されました。

ソリューション

Breyer 氏は、異なるハードウェア・アーキテクチャー、特に GPU を搭載したアーキテクチャーにおけるソフトウェア・パフォーマンスの移植性を向上することに注力してきました。

> 「この目標を達成するため、SYCL* の可能性を探ってきました。動作はするがパフォーマンスの低いコードを記述するのではなく、さまざまなプラットフォームで効率良く動作するコードを記述することに喜びを感じます。」

oneAPI イニシアチブを推進する業界全体のエコシステムと、オープンソース・コミュニティーを通じて利用可能なツールキットとライブラリーは、彼のプロジェクトの目標達成に有用なものであることが証明されました。oneAPI 業界イニシアチブは、開発者が CPU とアクセラレーター・アーキテクチャーでプログラミングを統一できるようにする協調エコシステムであり、アプリケーションの高速化、効率的な開発プロセスの実現、摩擦のない開発を可能にする革新的な手法の創造を目的としています。

過去のプロジェクトの積み重ね

Breyer 氏は学士論文のときから、CUDA* を使用して HPC コードの移植性を高めるアイデアを練ってきました。

> 「修士論文では CUDA* ではなく SYCL* を使用しました。さらに、MPI を用いたマルチ GPU 実行をサポートし、異なるベンダーの複数の GPU をサポートする局所性鋭敏型ハッシュ (LSH: Locality Sensitive Hashing) を用いて、パフォーマンスを移植可能な K 近傍法の実装を開発しました。このアプローチは非常にユニークです。また、hipSYCL のほかに、インテル® oneAPI DPC++/C++ コンパイラーと ComputeCpp* をサポートするようにコードを書き換えました。」

開発ノートと実現するためのテクノロジー

ベンダー固有のプログラミング・モデルに対して強力なソース互換性を持つ hipSYCL は、AMD* の CUDA* 組込み関数や最適化テンプレート・ライブラリーをサポートし、オープンソースの開発者の間で人気があります。また、SYCL* 言語を強化および拡張する新しい SYCL*拡張を試す有用なプラットフォームを提供します。

Breyer 氏は次のように述べています。

> 「現時点では、hipSYCL は 1 つのホストプロセスのみを使用するマルチ GPU をサポートしていません。そのため、MPI を使用してマルチ GPU 対応を実装する必要がありました (1 GPU に 1 MPI プロセス)。SYCL* 実装は、それぞれ微妙に挙動が異なることが分かりました。例えば、DPC++ コンパイラーと ComputeCpp* は、hipSYCL では問題なくコンパイルできるコードを拒否しました。」

異なるベンダーの GPU をサポートする Breyer 氏の研究「局所性鋭敏型ハッシュと SYCL* を使用した分散型 K 近傍法」(英語) は GitHub* で見ることができます。

Breyer 氏は、主な計算作業は GPU で行っていると説明しています。これには、ハッシュ関数やテーブルの作成、K 近傍法の計算などが含まれます。これらのタスクは、利用可能なすべての GPU に均等に割り当てられます。CPU ワークロードは、I/O 操作や MPI 通信などの補助的なタスクに集中します。

図 1 は、典型的な hipSYCL 実装に含まれるツールチェーンを示したものです。

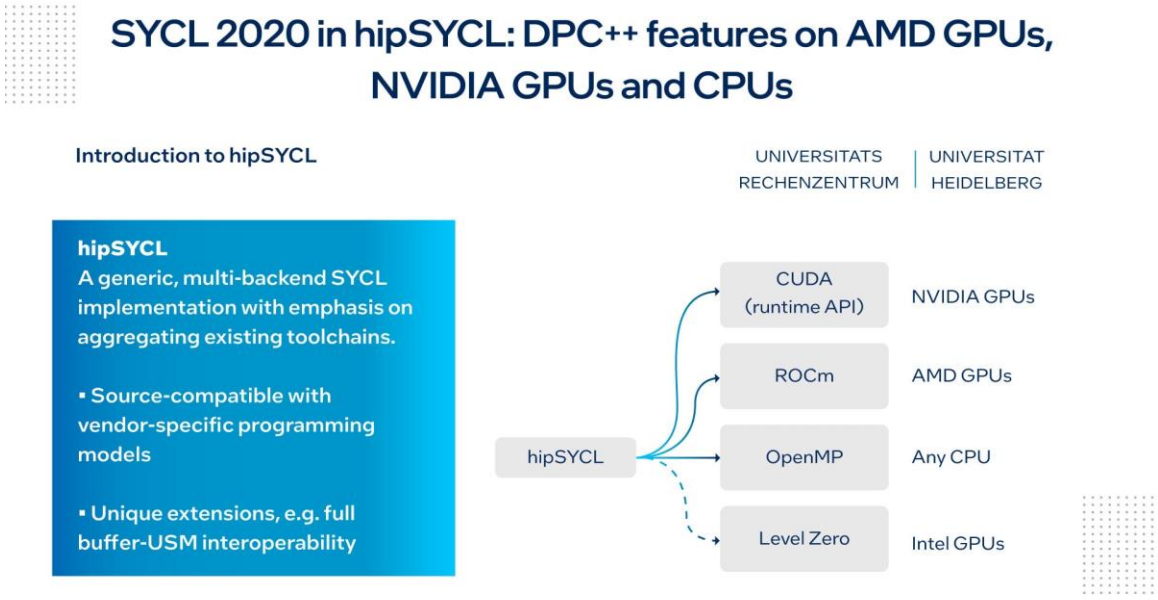

図 1. hipSYCL を使用してツールチェーンを集約

CUDA* コードから DPC++ コードへの移行を効率化するツールであるインテル® DPC++ 互換性ツール (図 2) は、有用なコンパイラー、ライブラリー、解析、デバッグツールを集めたインテル® oneAPI ベース・ツールキットの一部として提供されています。

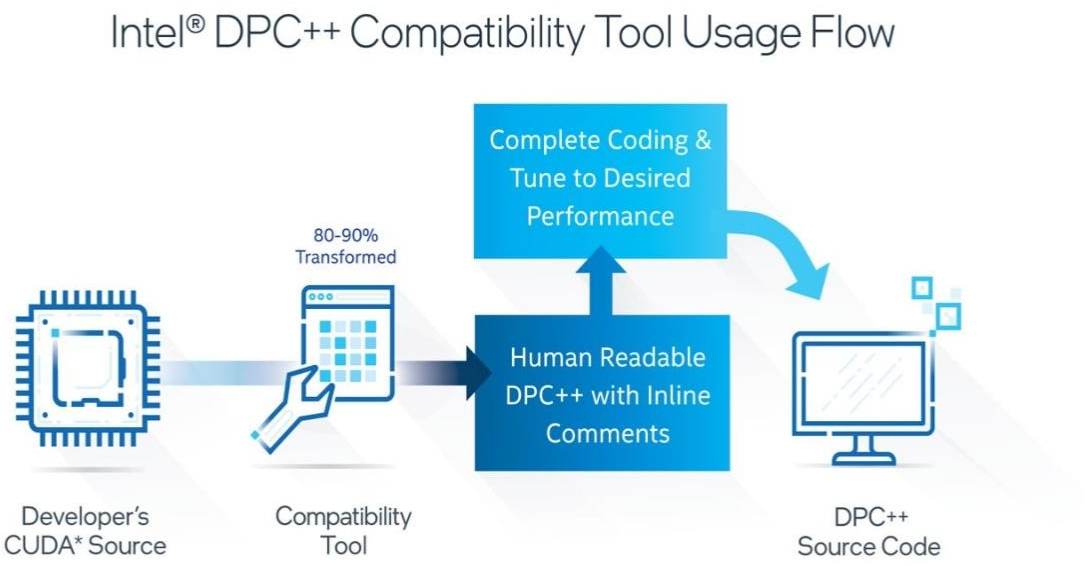

図 2. インテル® DPC++ 互換性ツールの使用フロー

ほかの開発者へのアドバイス

このプロジェクトで得た経験を基に、Breyer 氏はほかの開発者に向けて次のようにアドバイスしています。

> 「複数の SYCL* 実装をサポートする場合は、DPC++ コンパイラーで開発します。このコンパイラーでは、開発プロセスのできるだけ早い段階で潜在的なバグを発見できるように、コンパイル時のチェックが強化されています。」

Breyer 氏は、すべての SYCL* 実装のパフォーマンスはほぼ同じであり、SYCL* 1.2.1 のほぼ同じ機能セットを実装していると指摘した上で、次のように補足しています。

> 「ただし、3 つの SYCL* 実装すべてで CMake* を使用する場合は、CMake* バージョン 3.20.0 以降 (英語) を使用し、hipSYCL パッケージがサポートされていることを確認します。ComputeCpp* はインクルードして使用します。」

また、次のようにも述べています。

> 「DPC++ コンパイラーは、インテル® DevCloud にプリインストールされているので、簡単に開発を始めることができます。また、『Data Parallel C++: Mastering DPC++ for Programming of Heterogeneous Systems Using C++ and SYCL』は非常に優れたリソースであり、一読することをお勧めします。Open Access (英語) から PDF 版を無料でダウンロードできます。」

> 「複数のコンピュート・アーキテクチャーに単一のプログラミング環境を提供する oneAPI は、ヘテロジニアス・コンピューティングの可能性を解き放つ重要なツールです。科学コミュニティーは、複数のハードウェア・プラットフォームでのコード開発への投資を活用し、異なるハードウェア・ターゲットからのパフォーマンス向上を促進し、将来のハードウェア・ターゲットを利用しやすくします。」[1]
>
> ケンブリッジ大学リサーチ・コンピューティング・サービス担当ディレクター
> Paul Calleja 氏

まとめ: 継続中の研究

プロジェクトの主な成果を尋ねると、Breyer 氏は次のように答えました。

> 「異なるベンダーの GPU を搭載した異なるハードウェア・プラットフォームで、相応のパフォーマンスを達成しました。3 つの SYCL* 実装のすべてでパフォーマンスが向上しました。また、最大 8 台の NVIDIA* GTX 1080 Ti GPU で、ほぼ完璧な並列効率を実現しました。」

Breyer 氏は引き続き、以下の研究と取り組みを行っています。

- ユークリッド距離以外の距離メトリックへの対応
- 基本 LSH アルゴリズムのバリエーションの開発
- ほかの種類の LSH 関数の開発
- エントロピー・ベースのハッシュ関数を作成するより効率的な実装
- マルチノード・システムでのパフォーマンスの調査

Breyer 氏の画期的なテクノロジーの進歩は、hipSYCL や oneAPI を含むいくつかの面で進行中です。研究と発見は、ハイデルベルク大学コンピューティング・センターの oneAPI 研究拠点から発信されています。これらは、ヘテロジニアス・コンピューター・アーキテクチャー向けの効果的なオープン・プログラミング・モデルを構築するため、オープンソース・コミュニティー内の共同の世界的なイニシアチブの一部となっています。

インテル コーポレーションのデータセンター XPU 製品 & ソリューション担当バイスプレジデントの Jeff McVeigh は次のように述べています。

> 「oneAPI は、オープンで協調的なアプローチにより、さまざまなアーキテクチャーでコードの再利用を効率化し、多様なワークロードの開発を簡素化しようとする真の産業間イニシアチブです。URZ の研究は、先進的な DPC++ アプリケーションのサポートをほかのアーキテクチャーに拡大することで、oneAPI のクロスベンダーの実現に貢献しています。」[2]

リソースと推奨事項

局所性鋭敏型ハッシュと SYCL* を使用した分散型 K 近傍法 (英語)
詳細はインテル® DevMesh プロジェクト・ページを参照してください。

GitHub* の hipSYCL (英語)
実装のコードと長期的な可能性を探ることができます。

oneAPI 研究拠点 (英語)
oneAPI エコシステムの成長と繁栄の支援に関する情報が得られます。

インテル® DevCloud (英語)
このサイトでは、最先端のインテルの CPU、GPU、FPGA、およびツール、フレームワーク、ライブラリーを含むプリインストールされたインテル® oneAPI ツールキットに自宅から無料でアクセスしてコードを開発できます。登録 (英語) して数分で作業を開始できます。

[1] oneAPI とは? (英語)

[2] ハイデルベルグ大学コンピューティング・センター (URZ) が oneAPI 研究拠点を開設 (英語)。URZ Newsroom。2020 年 9 月。

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。