

# インテル® oneAPI ベース・ツールキットを使用した GPU アプリケーションの最適化

この記事は、[software.intel.com](https://software.intel.com) に公開されている「[Optimize Your GPU Application with Intel® oneAPI Base Toolkit](#)」の日本語参考訳です。

---

## ここで学べること

インテルの最新のディスクリート GPU と統合 GPU、およびインテル® oneAPI ソフトウェアを使用して、グラフィックス・プロセッシング・ユニット (GPU) を最適化する方法をステップごとに紹介します。

- 特に、データ並列プログラミングを容易にするため、ISO C++ をベースに SYCL\* 標準とコミュニティによる拡張を組み込んだ[データ並列 C++ \(DPC++\)](#) の使用方法に注目します。
- また、各セクションで紹介されているヒントや関連情報を参考にして、ニーズに合ったアプローチを採用できます。

[さまざまな oneAPI 計算ワークロードで CPU、GPU、FPGA の利点を比較](#)してみてください。

## 対象ユーザー

インテルの最新の統合および外付けグラフィックス・カードとクロスコードの再利用を可能にする DPC++ を使用して、コードを高速化することに興味があるソフトウェア開発者。独自の C++ または CUDA\* コードを使用するか、インテルの豊富なサンプル・アプリケーションを使用して開始できます。

インテル® GPU とインテル® oneAPI ベース・ツールキット・ソフトウェアが必要です。ローカルの開発システムを使用するか、無料のインテル® DevCloud 仮想サンドボックスを使用できます。インテル® DevCloud では、各種インテル® GPU とインテル® oneAPI ソフトウェア・ツールを利用できます。GPU の選択方法は、「[ステップ 1: GPU ハードウェアの選択](#)」を確認します。

## ワークフロー

[ステップ 1: GPU ハードウェアの選択](#)

[ステップ 2: サンプルコードの選択](#)

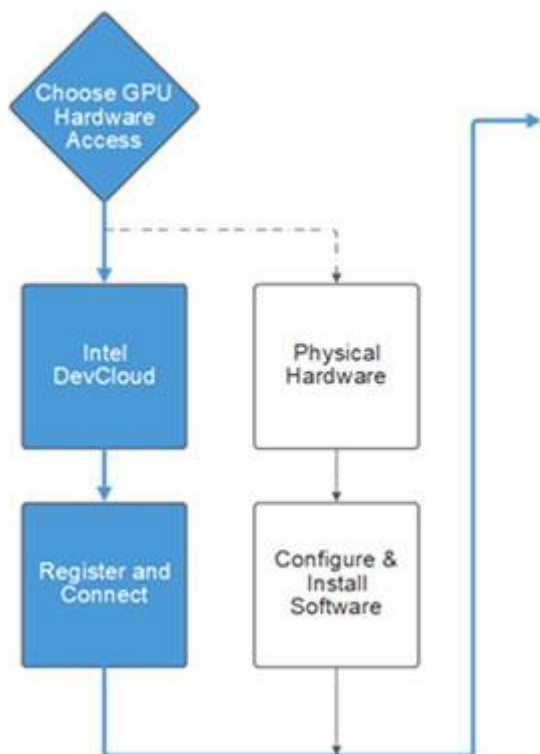
[ステップ 3: インテル® Advisor でオフロード可能なコード領域を調査](#)

[ステップ 4: インテル® oneAPI の各種コンパイラとライブラリーでコードをオフロードして最適化](#)

[ステップ 5: インテル® Advisor でオフロード効率を評価](#)

[ステップ 6: インテル® VTune™ プロファイラーでアプリケーション全体のパフォーマンスを確認](#)

## ステップ 1: GPU ハードウェアの選択



最新のインテル® アーキテクチャーを採用している場合でも、クラウドリソースを必要としている場合でも、最適化戦略はハードウェアから始まります。GPU オフロードを最大限に活用するには、どのようなハードウェア・リソース向けに最適化する必要があるのかを理解する必要があります。

### ハードウェアとソフトウェアの選択肢

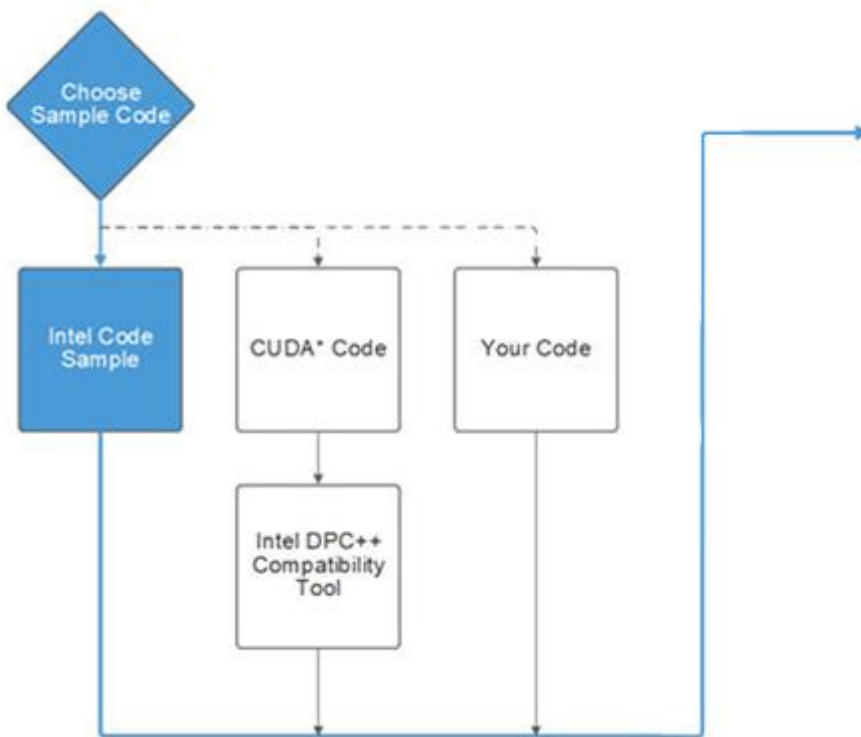
- クラウドベースの開発用サンドボックスである無料のインテル® DevCloud でインテルのハードウェアとソフトウェアを使用します。
- 所有しているインテル® グラフィックス・プロセッサと、無料のインテル® oneAPI ベース・ツールキットをインストールして使用します。

	インテル® DevCloud	インテル® グラフィックス・プロセッサ
入手方法	<p>インテル® DevCloud は、oneAPI クロスアーキテクチャー・アプリケーションについて学び、プログラミングするための無料のクラウドベースの開発サンドボックスです。</p> <p><b>ハードウェア</b> インテル® DevCloud では、新しいインテル® Iris® Xe MAX GPU などの実際のインテル® GPU ハードウェア上でコードを最適化できます。インテル® Xeon® スケーラブル・プロセッサと® FPGA も利用できます。</p> <p><b>ソフトウェア</b> インテル® DevCloud には、インテル® oneAPI ベース・ツールキットや HPC、AI、IoT 向けのアドオン・ツールキットなどのインテル® ソフトウェアがプリインストールされており、すぐに使用できます。</p> <p><a href="#">アクセスするためサインアップ (英語)</a> して、任意のオペレーティング・システムでインテル® DevCloud に接続します。</p>	<p>独自の開発環境で、インテル® グラフィックス・ハードウェアと無料のインテル® oneAPI ベース・ツールキットを使用します。</p> <p><b>ハードウェア</b> インテル初のディスクリット GPU であるインテル® Iris® Xe MAX グラフィックスを利用して、最新のインテル® ハードウェア上でオフロードと最適化の戦略を調査します。<a href="#">インテル® Iris® Xe MAX グラフィックスの詳細 (英語)</a> 統合 GPU を含む多くのインテル® プラットフォームを基本的な GPU オフロードの概念実証と可能性の検証に使用できます。 <a href="#">適切な GPU アーキテクチャーを見つける (英語)</a></p> <p><b>ソフトウェア</b> インテル® oneAPI ベース・ツールキットは、多様なアーキテクチャー向けにハイパフォーマンスなアプリケーションを開発するのに必要なツールとライブラリーの基本セットを提供します。 <a href="#">無料のインテル® oneAPI ベース・ツールキットをダウンロードする (英語)</a></p>
OS	<p>ホスト: Windows*, Linux*, macOS* インテル® DevCloud (リモート): Linux*</p>	Linux*

	インテル® DevCloud	インテル® グラフィックス・プロセッサ
ソフトウェア	インテル® oneAPI ベース・ツールキット (インテル® DevCloud にプリインストール)	インテル® oneAPI ベース・ツールキット (英語)
GPU	利用可能な GPU: インテル® Iris® X® MAX GPU (インテル® DevCloud でノードを利用可能)	サポートされる GPU: <ul style="list-style-type: none"> <li>• インテル® Iris® X® MAX GPU</li> <li>• インテル® Iris® X® グラフィックス (統合 GPU)</li> <li>• インテル® UHD グラフィックス</li> </ul>
言語	DPC++	DPC++
インターフェイス	コマンドライン・インターフェイス (CLI)	コマンドライン・インターフェイス (CLI)

**ヒント:** インテル® Advisor のオフロードのモデル化機能は、さまざまなハードウェア・ソリューションにおけるパフォーマンス向上と投資対効果 (ROI) を予測することで、ニーズに合った最適なハードウェアを選択するのに役立ちます。

## ステップ 2: サンプルコードの選択



インテルのサンプル、既存の CUDA\* ソースコード、または独自の C++ アプリケーションから開始できます。

### インテル® oneAPI サンプルコードを使用する

インテル® oneAPI ベース・ツールキットには、DPC++ を使用して GPU 上で並列処理を表現するためのさまざまな手法を示すサンプルのコレクションが含まれています。ISO3DFD サンプルを使用して、DPC++ と GPU 最適化を開始できます。この計算集約型の OpenMP\* サンプルは、3次元媒体での音響等方性波動伝播のシミュレーションを行います。ISO3DFD サンプルを使用して、DPC++ を使用した最適化

戦略に利用できるさまざまなツールやリソースを段階的に確認できます。

GitHub\* からインテル® DevCloud または開発環境へサンプルをダウンロードします。サンプルのビルドと実行の詳しい手順は、サンプルの README にあります。

[GitHub\\* 上の ISO3DFD サンプル \(英語\)](#)

[GitHub\\* 上のすべての oneAPI サンプル \(英語\)](#)

## 関連情報

- コマンドラインからサンプルを参照してダウンロード: [ガイド](#) | [ビデオ](#) (英語)
- Eclipse\* からサンプルを参照: [ガイド](#) | [ビデオ](#) (英語)
- Visual Studio\* Code 拡張からサンプルを参照: [ガイド](#)

## 独自のコードを使用する

### 既存の CUDA\* コードの移行

既存の CUDA\* コードを DPC++ のマルチプラットフォーム・コードに移行します。インテル® DPC++ 互換性ツールは、CUDA\* 言語カーネルとライブラリー API 呼び出しの両方を DPC++ に移行し、80%-90% の CUDA\* コードをアーキテクチャーとベンダーに移植可能な DPC++ コードへ自動的に移行します。インラインコメントは、DPC++ コードの記述やチューニングに役立ちます。サンプル CUDA\* プロジェクトも用意されており、移行プロセス全体の参考になります。

1. [インテル® DPC++ 互換性ツール](#)を使用して CUDA\* コードを移行します。
2. [ステップ 3](#) に進み、アプリケーションをビルドして、オフロードのモデル化機能で DPC++ コードを評価して、さらなるオフロードの可能性を探ります。

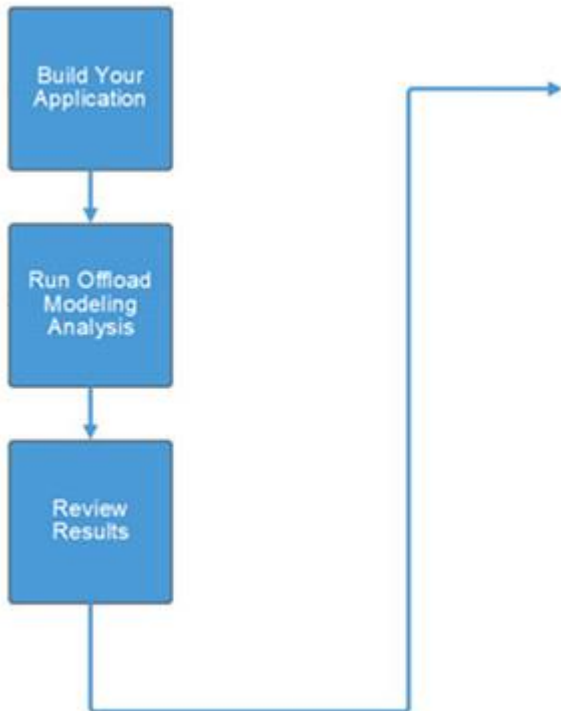
### 独自の C++ プロジェクトの最適化

独自の C++ コードを使用するには、開発環境を設定して、このワークフローの手順に従います。ここで紹介する最適化手法は、既存のプロジェクトに直接適用できます。

- C++ アプリケーションをインテル® DevCloud にコピーします。  
または
- 独自の開発環境を使用します。

次に、[ステップ 3](#) に進み、アプリケーションをビルドして、オフロードのモデル化機能で DPC++ コードを評価して、さらなるオフロードの可能性を探ります。

## ステップ 3: インテル® Advisor でオフロード可能なコード領域を調査



インテル® Advisor ツールはコードを解析して、GPU オフロードに最適な候補を特定するのに役立ちます。オフロードのモデル化機能は、パフォーマンス・スピードアップの予測、オフロード・オーバーヘッドの推定、パフォーマンス・ボトルネックの特定を行います。異なるハードウェア・ソリューションをモデル化してパフォーマンスを最大化することで、ROI を向上できます。

### オフロードのモデル化解析を実行する

インテル® Advisor は、関数内のデータの移動、メモリー・アクセス・パターン、計算量を測定して、インテル® GPU 上でのコードのパフォーマンスを予測します。恩恵を受ける可能性が最も高いコード領域を、オフロードの最初のターゲットにするべきです。

次の手順に従って、オフロードのモデル化を実行します。

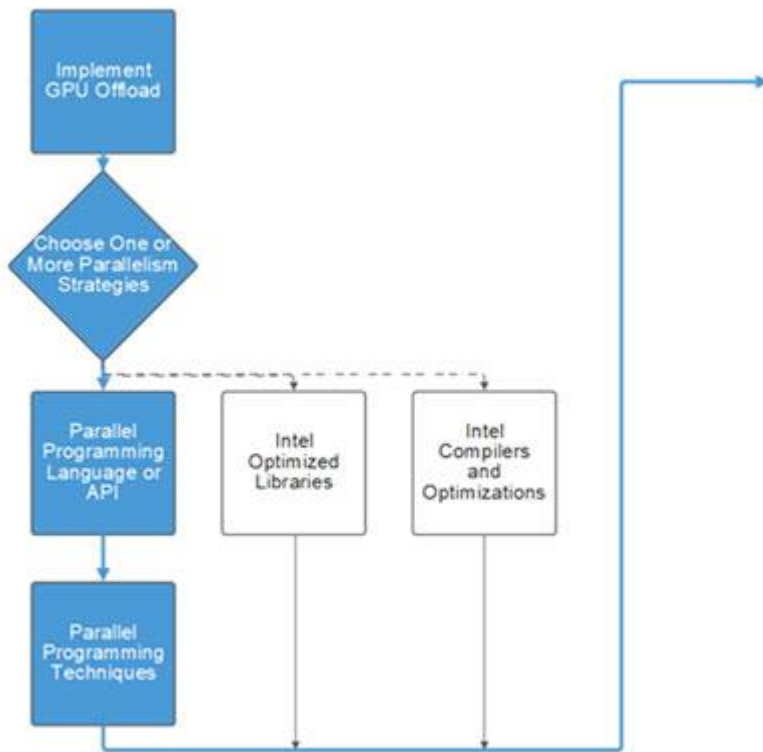
1. [適切な環境変数を使用してサンプル・アプリケーションをビルド](#) (英語) します (これは、DPC++、OpenMP\*、および OpenCL\* で必要です)。
2. [オフロードのモデル化解析を実行](#) (英語) します。
3. [結果を確認](#) (英語) します。

**ヒント:** インテル® Advisor には、プロジェクトの作成や解析を実行するためのグラフィカル・ユーザー・インターフェイスもあります。

### 関連情報

- [ターゲット GPU での C++ アプリケーションのスピードアップを予測](#)
- [GPU にオフロードするコード領域を特定して GPU の使用状況を可視化](#)
- [ビデオ: インテル® Advisor を使用して効率良く GPU へオフロードする](#) (英語)

## ステップ 4: インテル® oneAPI の各種コンパイラーとライブラリーでコードをオフロードして最適化



アプリケーションのニーズ、インテル® Advisor のアドバイス、利用可能なハードウェアに応じて、最良の最適化戦略を選択してコードを変更します。ドキュメント、サンプル、トレーニングは、パフォーマンスを最大化するデザイン決定に役立ちます。

### GPU オフロードの実装

最初に、インテル® Advisor の結果を基に、推奨されたコードを GPU デバイスへオフロードします。次に、複数の手法を組み合わせて最適化戦略を立てます。

### 基本的な DPC++ フレームワーク

DPC++ アプリケーションを記述します。

1. デバイスを選択します。
2. デバイスキューを宣言します。
3. バッファを宣言します。
4. ジョブを送信します。

### 並列化戦略を立てる

oneAPI では、以下を組み合わせて並列化戦略を立てることを推奨しています。

- インテルにより最適化されたライブラリー: 『[インテル® oneAPI プログラミング・ガイド](#)』の「API ベースのプログラミング」
- インテル® コンパイラーと最適化: 『[インテル® oneAPI DPC++/C++ コンパイラー・デベロッパー・ガイドおよびリファレンス](#)』
- 並列プログラミング言語や API: 『[インテル® oneAPI プログラミング・ガイド](#)』の「oneAPI プログラミング・モデル」>「データ並列 C++ (DPC++)」

## 関連情報

- [DPC++ 言語ガイド \(英語\)](#)
- 書籍: 『[Data Parallel C++: Mastering DPC++ for Programming of Heterogeneous Systems Using C++ and SYCL\\*](#) (データ並列 C++: C++ と SYCL\* を使用したヘテロジニアス・システムのプログラミング向けに DPC++ をマスターする)』(英語)
- [インテルのサンプルを使用して DPC++ を開始する](#)
- [インテル® DevCloud 上のトレーニング: データ並列 C++ の要点 \(英語\)](#)

## GPU オフロードの最適化

『[インテル® oneAPI GPU 最適化ガイド](#)』を使用して GPU の最適化戦略を立てます。

### オフロードの最適化の基本

以下を考慮します。

- アムダールの法則
- 局所性の重要性
- 適切なワークサイズ

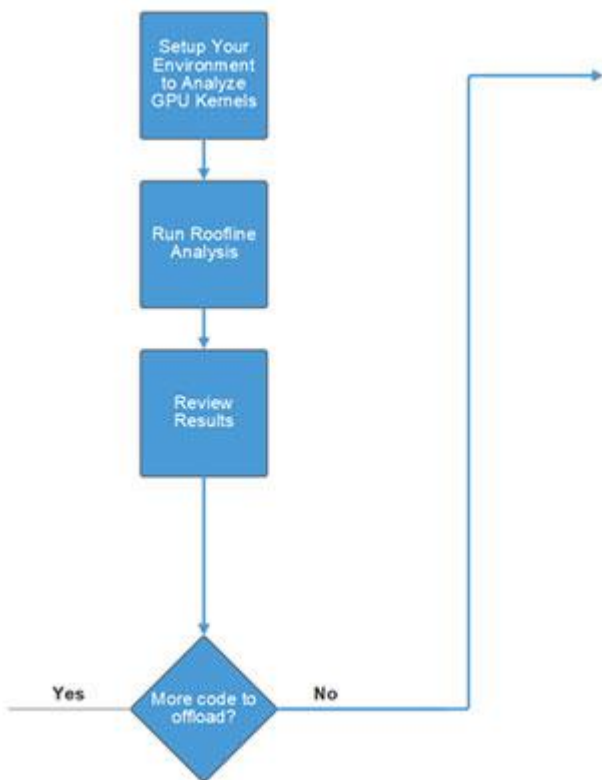
詳細は、『[インテル® oneAPI GPU 最適化ガイド](#)』の「導入ガイド」を参照してください。

### 最適化の次のステップ

最適化ワークフローの主なフェーズ (『[インテル® oneAPI GPU 最適化ガイド](#)』の「はじめに」 > 「最適化ワークフロー・フェーズ」を参照):

- CPU と GPU のオーバーラップを理解する
- すべての計算リソースをビジー状態に保つ
- ホストとデバイス間の同期を最小限にする
- ホストとデバイス間のデータ転送を最小限にする
- データを高速メモリーに保持し、適切なアクセスパターンを使用する

## ステップ 5: インテル® Advisor でオフロード効率を評価



アプリケーションを変更したら、Intel® Advisor の GPU ルーフライン解析を使用して、オフロードされたコードの実際のパフォーマンスを測定します。Intel® Advisor は、ベンチマークとハードウェア・メトリックのプロファイルを使用して GPU カーネルのパフォーマンスを測定します。制限を明らかにし、さらなる最適化により最も大きな恩恵が得られるコード領域を特定します。

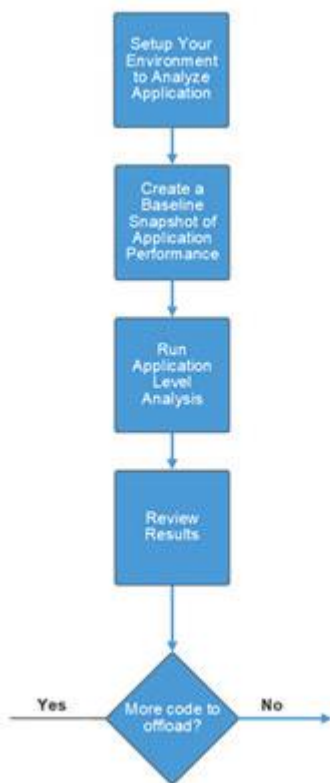
### GPU ルーフライン解析を実行してオフロードコードを変更する

GPU コードを評価して、パフォーマンスがハードウェアの上限にどれだけ近いかを確認します。

1. GPU カーネルを解析するため環境を設定(英語) します。
2. ルーフライン解析を実行 (英語) します。
3. 結果を確認して、ハードウェア・モデルに基づいてスループットを評価 (英語) します。
4. ボトルネックが見つかった場合は、「ステップ 4: オフロードと最適化」に戻り、問題に対応するためコードを変更します。



## ステップ 6: インテル® VTune™ プロファイラーでアプリケーション全体のパフォーマンスを確認



GPU オフロードコードを最適化したら、インテル® VTune™ プロファイラーを使用して、すべてのデバイスでアプリケーション全体のパフォーマンスを最適化します。インテル® VTune™ プロファイラーの解析結果は、最適化のためのガイダンスを提供します。

**ヒント:** インテル® VTune™ プロファイラーには、プロジェクトの作成や解析を実行するためのグラフィカル・ユーザー・インターフェイスもあります。

### アプリケーション・パフォーマンスのベースライン・スナップショットを作成する

パフォーマンス・スナップショットを使用して、アプリケーション・パフォーマンスのベースラインを作成し、詳しく解析する領域を特定します。

1. [GPU 解析用にシステムを設定](#) (英語) します。
2. [インテル® VTune™ プロファイラーのコマンドライン・インターフェイスを起動](#) (英語) します。
3. [パフォーマンス・スナップショット解析を実行](#) (英語) します。
4. 結果を表示します。

- インテル® DevCloud: [サマリーレポートを表示](#) (英語) します。
- インテル® DevCloud + ローカルにインストールされたインテル® VTune™ プロファイラー: 結果をローカルシステムにコピーして、[プロジェクトを作成](#) (英語) し、[インテル® VTune™ プロファイラーにインポート](#) (英語) します。
- インテル® oneAPI ベース・ツールキット: [インテル® VTune™ プロファイラーで結果を表示](#) (英語) します。

### アプリケーションが CPU 依存または GPU 依存かを評価します。

ホストとデバイス間の転送操作にかかった時間を確認して、CPU の最適化を開始します。次に、GPU が効率良く利用されていない領域を特定して、GPU を最適化します。

1. [GPU オフロード解析を実行](#) (英語) します。
2. アプリケーションの CPU パフォーマンスを最適化します。
3. [GPU 計算/メディア・ホットスポット解析を実行](#) (英語) します。
4. 「[ステップ 4: オフロードと最適化](#)」に戻り、アプリケーションの GPU パフォーマンスをさらに向上します。

### 関連情報

- [DPC++ のプロファイル](#)
- [インテル® VTune™ プロファイラーを使用してインテル® GPU 向けにアプリケーションを最適化](#)

## 製品および性能に関する情報

<sup>1</sup> 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/](http://www.Intel.com/PerformanceIndex/) (英語) を参照してください。