https://techdecoded.intel.io/essentials/profile-dpc-and-gpu-workload-performance/

intel Software

DPC++ と GPU ワークロードのパフォーマンスをプロファイル

インテル® VTune™ プロファイラー

インテル コーポレーション
テクニカル・コンサルティング・エンジニア
Vladimir Tsymbal

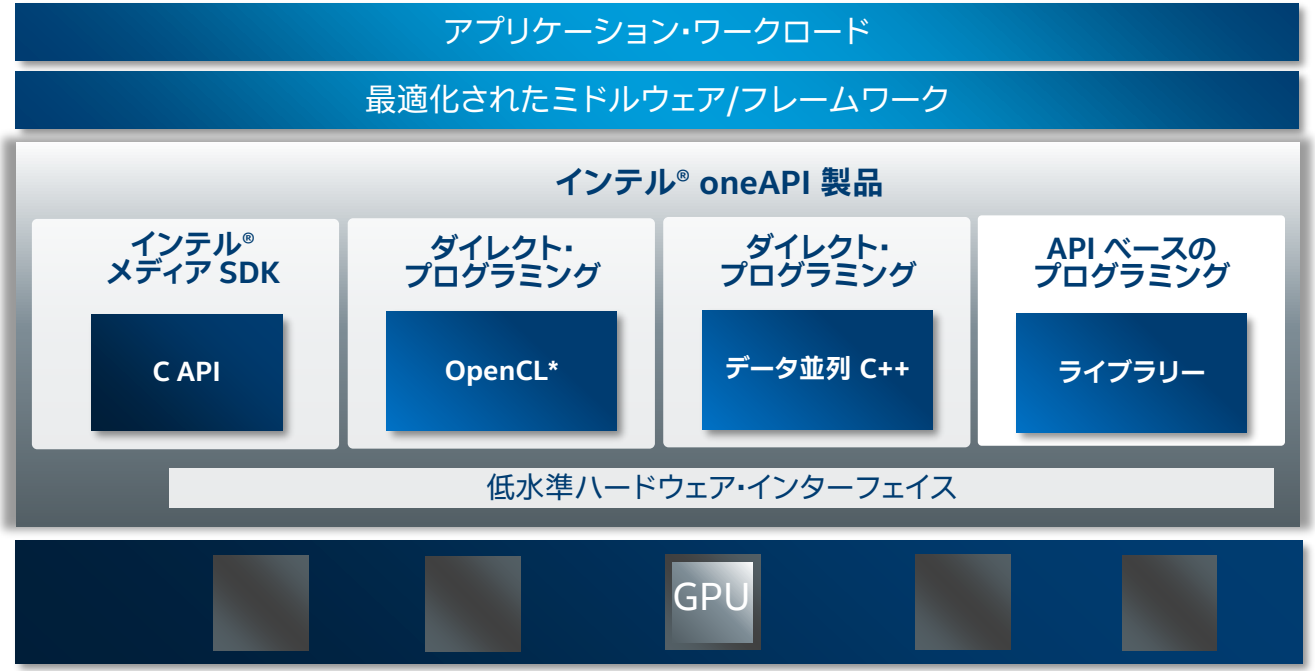
内容

- GPU プログラミング・モデルの概要
- インテル® VTune™ プロファイラーの GPU 解析の概要
- オフロード・パフォーマンスのチューニング
- GPU 計算/メディア・ホットスポット
- まとめ

© 2021 Intel Corporation.

インテルの GPU とプログラミング・モデル

- Gen9
  - モバイル、デスクトップ、ワークステーションで一般的
- Gen11
  - Ice Lake† CPU 搭載のモバイル・プラットフォーム
- Gen12
  - Tiger Lake† CPU 搭載のモバイル、Rocket Lake† CPU 搭載のデスクトップ



詳細は、software.intel.com/oneapi (英語) を参照

† 開発コード名

© 2021 Intel Corporation.

インテル® VTune™ プロファイラーの GPU 解析の概要

GPU アプリケーション解析

GPU 計算/メディア・ホットスポット

- ホストと GPU を可視化
- HW イベントベースのパフォーマンス・チューニング手法
- タイムライン表示と集計表示を提供

GPU インカーネル・プロファイル

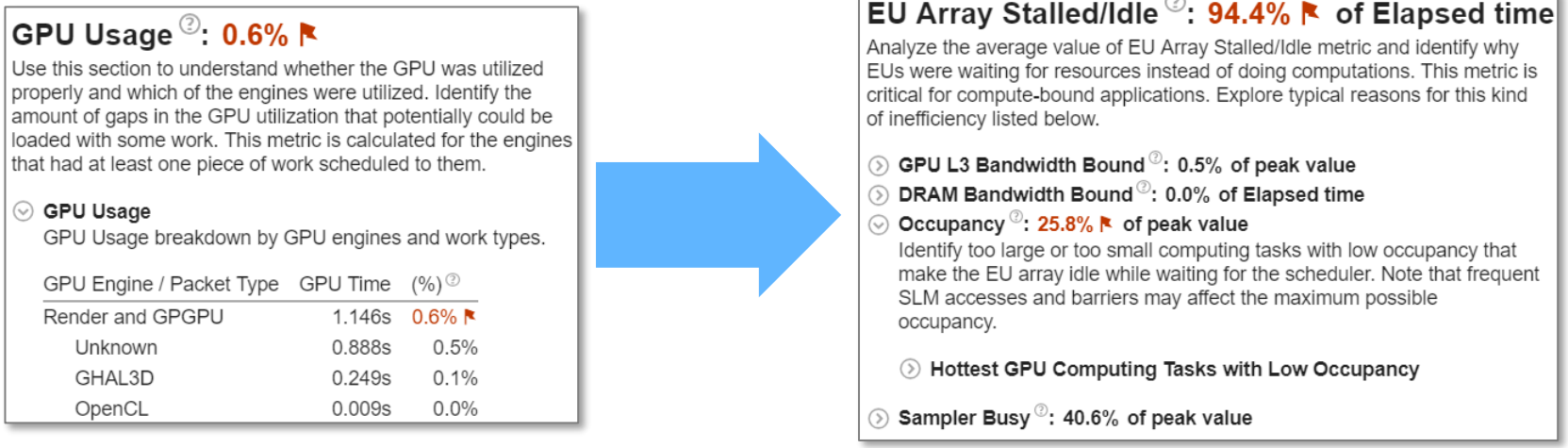
- ソース/命令レベルの GPU プロファイル
- SW インストルメンテーション
- 2 つのモード: 基本ブロック・レイテンシーとメモリー・アクセス・レイテンシー

GPU 占有率とプロファイルするカーネルを特定して、細粒度でカーネルをチューニング

© 2021 Intel Corporation.

GPU 解析: ホットスポット

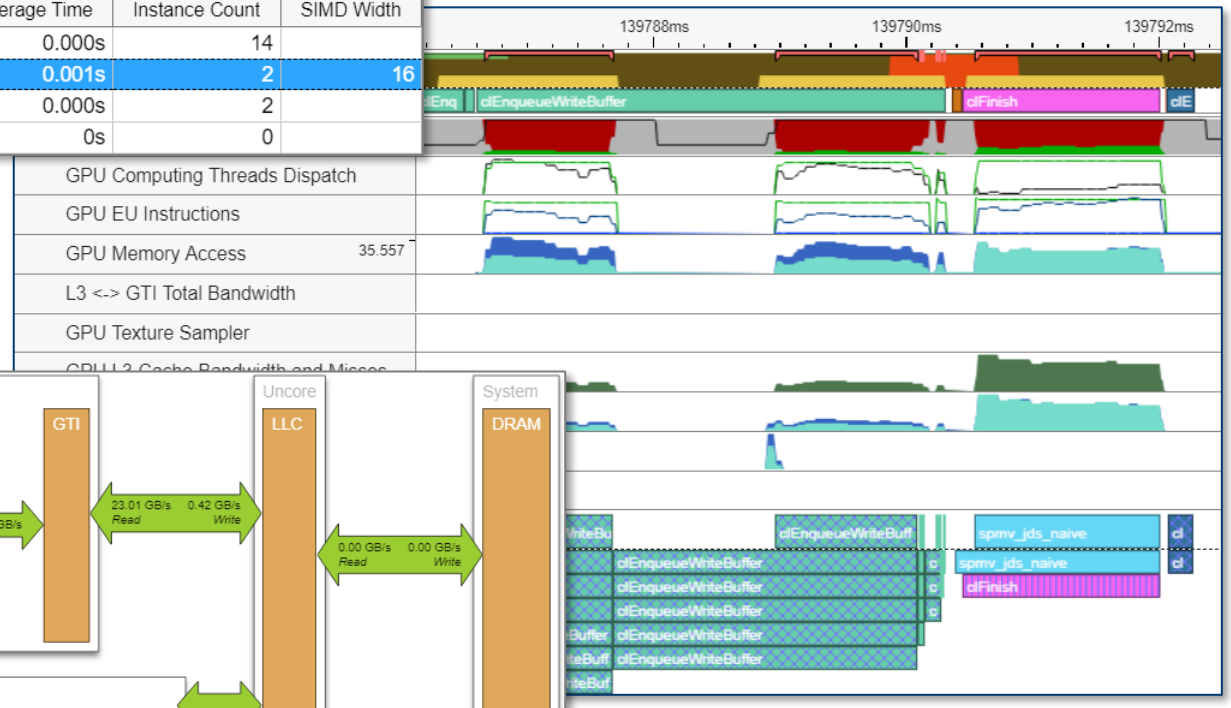
- [Summary (サマリー)] ページの各セクションの説明に従うだけ
- HW メトリックに基づくチューニング手法



© 2021 Intel Corporation.

GPU 解析: 集計表示とタイムライン表示

Computing Task	Work Size		Computing Task			
	Global	Local	Total Time ▼	Average Time	Instance Count	SIMD Width
clEnqueueWriteBuffer			0.005s	0.000s	14	
spmv_jds_naive	146944	256	0.003s	0.001s	2	16
clEnqueueReadBuffer			0.000s	0.000s	2	
[Outside any task]			0s	0s	0	

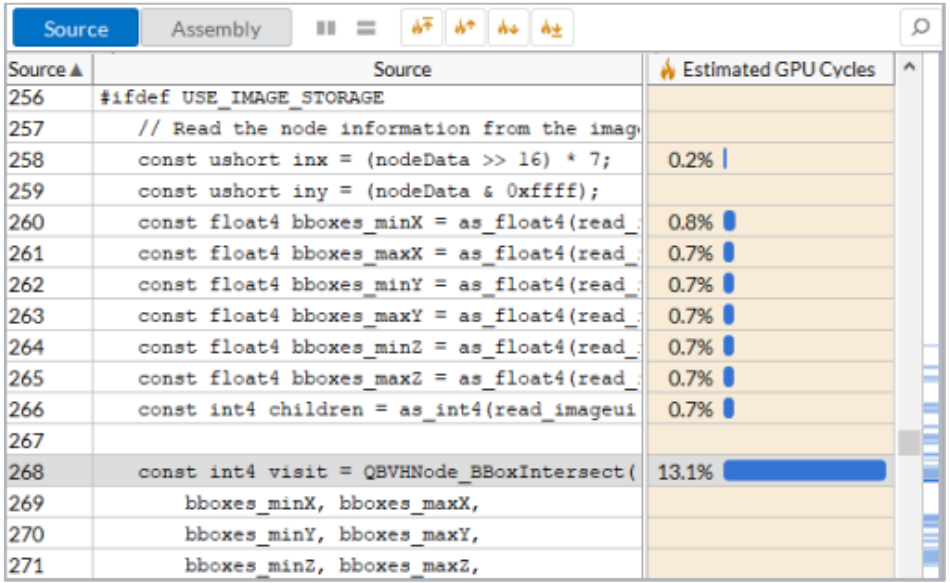


© 2021 Intel Corporation.

GPU 解析: インカーネル・プロファイル

GPU カーネル実行の解析

- メモリー・レイテンシーや非効率的なカーネル・アルゴリズムを検出
- OpenCL* ソースコードとアセンブリー・コードのホットスポットを確認
- DMA パケット実行を解析
  - パケットキューの深さヒストグラム
  - パケットの持続時間ヒストグラム
- GPU 側のコールスタック



© 2021 Intel Corporation.

オフロード･パフォーマンスのチューニング

GPU オフロード解析

ヘテロジニアス・アプリケーション

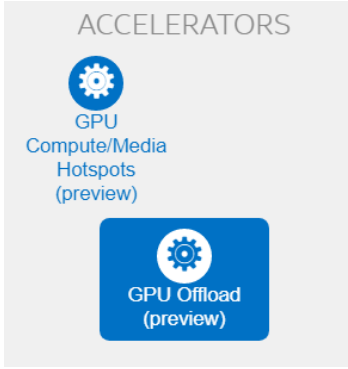
- オフロードモデル: SYCL*、DPC++、OpenMP*

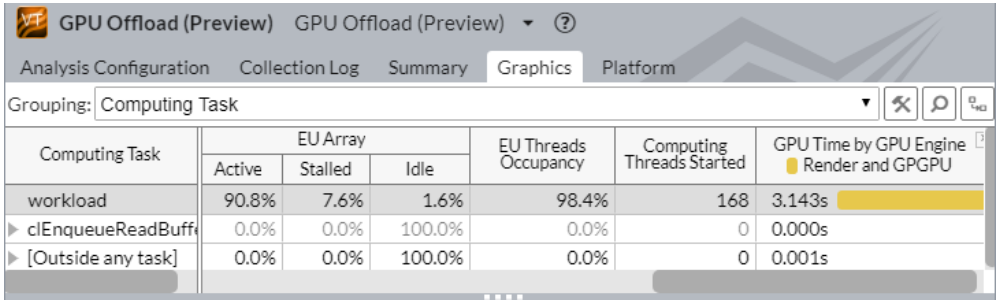
すべての実行リソースに注目

- 各種 CPU および GPU コア上のコード実行を調査
- CPU と GPU アクティビティーを関連付け
- アプリケーションが GPU 依存か、CPU 依存かを特定

詳しく解析するカーネルを特定

- タスクレベルの解析
- カーネルの効率
- データ転送レート





© 2021 Intel Corporation.

SYCL* について

SYCL* はホスト (CPU) のコードにデバイス (GPU など) のコードをインラインで埋め込む共有ソースデザインを実装

- OpenCL* を使用するアルゴリズムの再利用可能なテンプレート
- ホストとデバイスのコード間のタイプセーフな相互作用

言語拡張 (OpenCL*) を使用しないシングルソース C++

- OpenCL* と C++14 標準ベース

標準 C++ コンパイラーでコンパイルして、OpenCL* デバイスが利用できない場合は CPU にフォールバック

DPC++ = ISO C++ + Khronos SYCL* + コミュニティー拡張

© 2021 Intel Corporation.

DPC++ サンプル・アプリケーション

DPC++ は次のような追加機能を提供する SYCL* 拡張

- 統合共有メモリー (USM)
- ND-range サブグループ
- 順序付きキュー、など

GitHub* の oneapi-src/oneAPI-samples リポジトリーからさまざまなサンプル・アプリケーションを利用可能 (https://github.com/oneapi-src/oneAPI-samples (英語))

- 例: matrix_multiply_vtune
- 複数のカーネルを選択して行列乗算を実行
- 完全なオフロードの例ではない

© 2021 Intel Corporation.

https://github.com/oneapi-src/oneAPI-samples

```
void multiply1(int msize, int tidx, int numt, TYPE a[][NUM], TYPE b[][NUM], TYPE c[][NUM], TYPE t[][NUM]) {
  int i, j, k;

  default_selector device;
  queue q(device, exception_handler);

  range<2> matrix_range{NUM, NUM};

  buffer<TYPE, 2> bufferA((TYPE*)a, matrix_range);
  buffer<TYPE, 2> bufferB((TYPE*)b, matrix_range);
  buffer<TYPE, 2> bufferC((TYPE*)c, matrix_range);

  q.submit([&](cl::sycl::handler& h) {
    auto accessorA = bufferA.get_access<sycl_read>(h);
    auto accessorB = bufferB.get_access<sycl_read>(h);
    auto accessorC = bufferC.get_access<sycl_read_write>(h);

    h.parallel_for<class Matrix1<TYPE> >(matrix_range,[=](cl::sycl::id<2> ind) {
      int k;
      for (k = 0; k < NUM; k++) {
        accessorC[ind[0]][ind[1]] += accessorA[ind[0]][k] * accessorB[k][ind[1]];
      }
    });
  }).wait_and_throw();
}
```

deviceQueue を宣言

2 次元の範囲を宣言

3 つのバッファーを宣言して初期化

ジョブをキューに送信

バッファーへの 3 つのアクセサーを宣言 2RD, 1 WR.

matrix_range に対して行列乗算を並列に実行

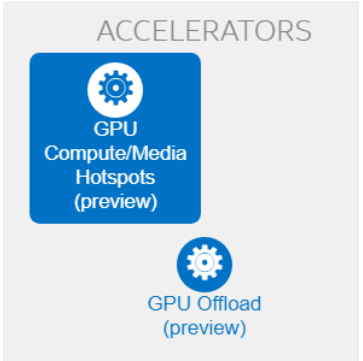
ind はこの範囲のインデックス

計算を実行: ind[0] は行、ind[1] は列

© 2021 Intel Corporation.

GPU 計算/メディア・ホットスポット

GPU 計算/メディア・ホットスポット



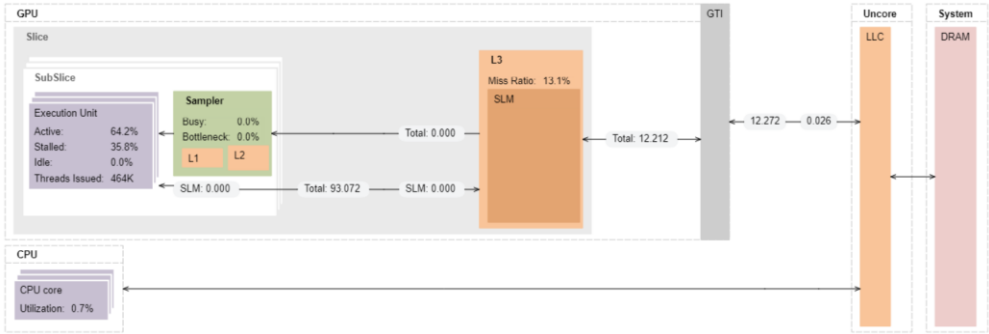
GPU 依存解析

- SoC 関連の一部のメトリックも測定される

GPU のパフォーマンスを最大限に引き出す方法

- 理想的には、最適化された IP ブロックを使用する - パフォーマンス・ライブラリー

- DPC++ のような高レベルのモデルでは GPU アーキテクチャーに合わせてワークロードの調整が可能
- インテル® VTune™ プロファイラーは HW レベルのメトリックを提供するため GPU ブロックの知識が必要



© 2021 Intel Corporation.

GPU アーキテクチャー

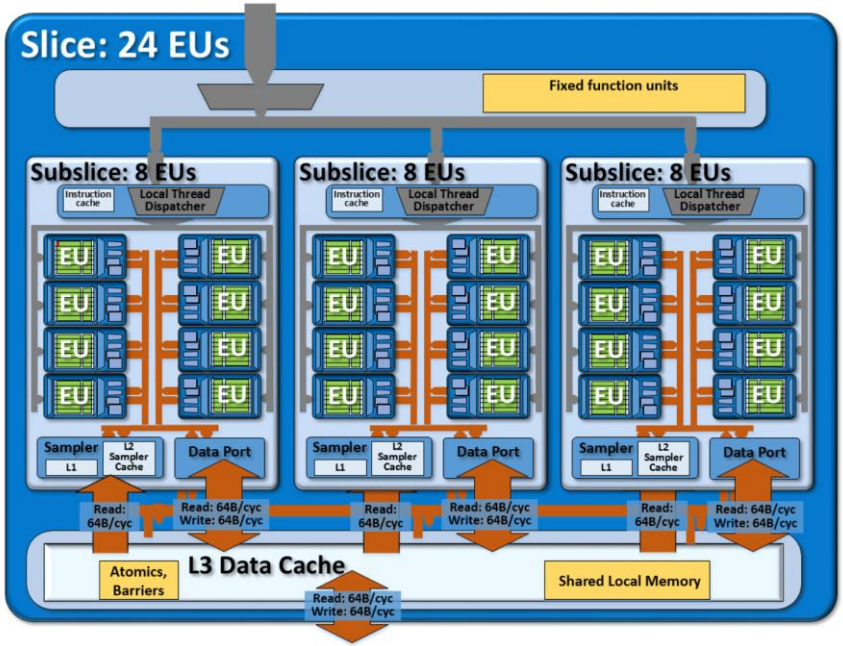
Collection and Platform Info

GPU

Name:	Intel(R) UHD Graphics 620
Vendor:	Intel Corporation
Driver:	27.20.100.8187
EU Count:	24
Max EU Thread Count:	7
Max Core Frequency:	1.1 GHz

GPU OpenCL Info

Version:	OpenCL C 2.0
Max Compute Units:	24
Max Work Group Size:	256
Local Memory:	64 KB
SVM Capabilities:	Fine-grained buffer with atomics



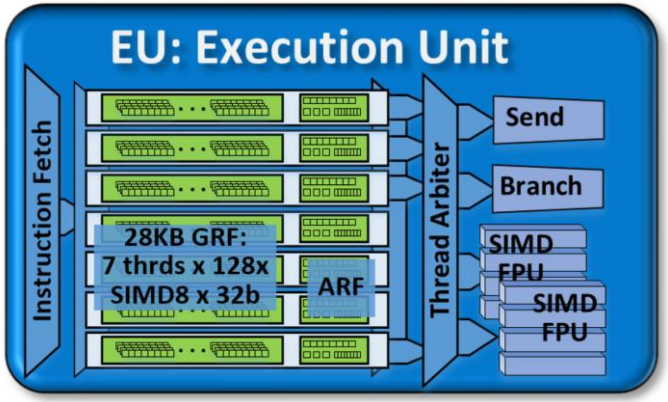
インテル® VTune™ プロファイラーの [Summary (サマリー)] ページから
GPU アーキテクチャーの詳細を素早く確認できる

© 2021 Intel Corporation.

Gen9 GPU の EU の詳細

EU の計算アーキテクチャー:

- 24 EU x 7 スレッド = 168 スレッドが利用可能
- 128 GRF (32 バイト、8 つの 32 ビット・データ要素のベクトルとしてアクセス可能)、柔軟なアクセスが可能
- 2 SIMD-4 FPU (32 ビットの FP または INT データ)
- 16 MAD/サイクル—(ADD + MUL) x 2 FPU x SIMD-4
- 2 つの追加ユニット: 分岐と送信



EU を最大限に活用することが目標

© 2021 Intel Corporation.

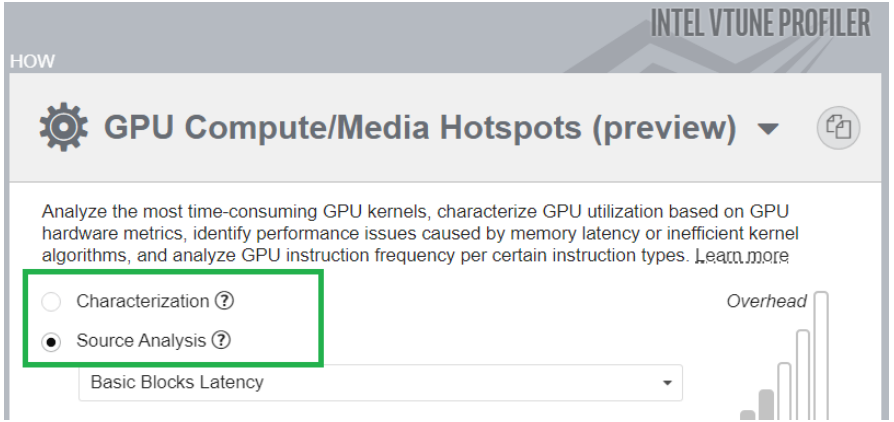
インテル® VTune™ プロファイラーの GPU 計算/メディア・ホットスポット解析

次のいずれかの GPU 解析設定を選択

- **[Characterization (特性化)]** – GPU エンジンの使用状況、効率、ストールを監視
- **[Source Analysis (ソース解析)]** – GPU カーネル内のパフォーマンスに重大な影響を与えるブロックとメモリーアクセスの問題を特定

最適化の方針:

- EU を最大限に有効活用
- SIMD を最大限に活用
- メモリーの問題による EU ストールを最小限に抑える



© 2021 Intel Corporation.

EU 効率とメモリー問題の解析

[Characterization (特性化)] オプションを使用

- EU アクティビティー: EU アレイアクティブ、EU アレイストール、EU アレイアイドル、開始された計算スレッド、コア周波数

[Overview (概要)] または [Compute Basic (基本計算)] メトリックを選択

- 追加のメトリック: メモリー読み取り/書き込み帯域幅、GPU L3 ミス、型付きメモリー読み取り/書き込みトランザクション

Grouping: Computing Task

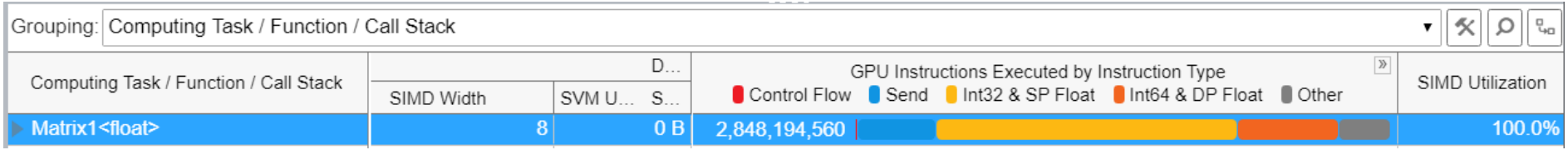
Computing Task	EU Array			EU Threads Occupancy	Computing Threads Started	EU Instructions			L3 Bandwidth, GB/sec
	Active	Stalled	Idle			IPC Rate	2 FPUs active	Send active	
Matrix1<float>	62.6%	37.4%	0.0%	95.1%	131,040	1.583	34.9%	6.8%	89.763

© 2021 Intel Corporation.

GPU 命令実行の解析

[Dynamic Instruction Count (動的命令数)] プリセットを使用

- カーネルで実行された命令の内訳
- 命令グループ
  - 制御フロー
  - 送信
  - 同期
  - Int16 & HP Float | Int32 & SP Float | Int64 & DP Float
  - その他

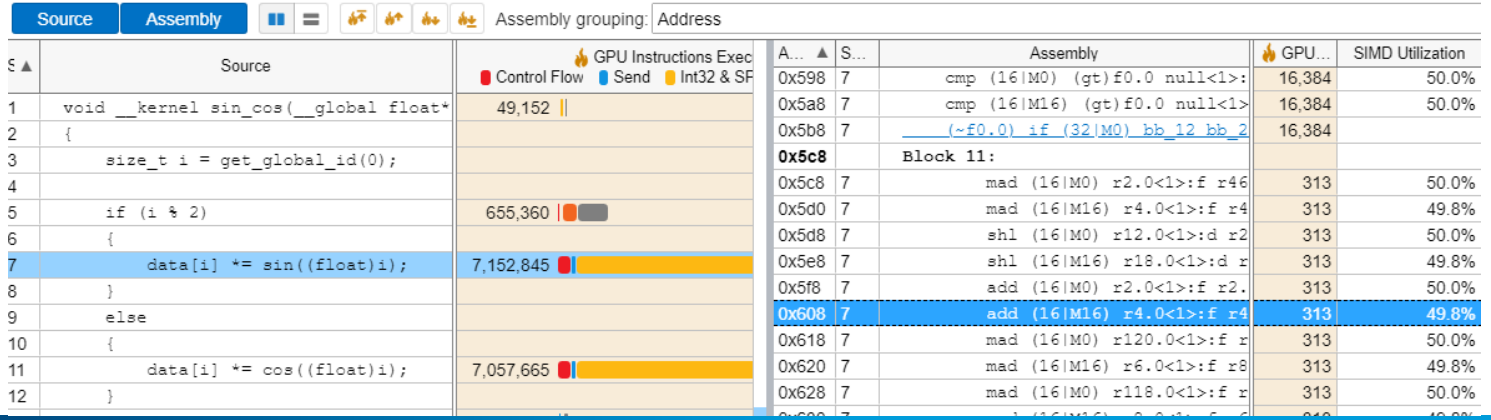


© 2021 Intel Corporation.

ソースコードの解析

[Source Analysis (ソース解析)] オプションを使用

- 注目するカーネルの基本ブロック・レイテンシーやメモリー・レイテンシーの問題を解析
- **[Source (ソース)]** と **[Assembly (アセンブリー)]** ペインを並べて表示



© 2021 Intel Corporation.

まとめ

- インテル® VTune™ プロファイラーは強力な GPU パフォーマンス解析ツール
- 新しいオフロードモデル SYCL*/DPC++/OpenMP* をサポート
- GPU パフォーマンスのチューニングを支援する 2 つのプレビュー解析タイプ: GPU オフロードと GPU 計算/メディア・ホットスポット
- タスク・スケジュールからアセンブリー命令レイテンシーまで、さまざまなパフォーマンス・メトリックにより、高レベルの推定値と低レベルの測定値を提供

© 2021 Intel Corporation.

関連情報

インテル® VTune™ プロファイラー
https://www.xlsoft.com/jp/products/intel/vtune/index.html

GPU オフロード解析:
https://www.isus.jp/wp-content/uploads/vtune/2020/help/index.htm

GPU 計算/メディア・スポット解析:
https://www.isus.jp/wp-content/uploads/vtune/2020/help/index.htm

DPC++ 仕様:
https://spec.oneapi.com/versions/latest/elements/dpcpp/source/index.html (英語)

GitHub* で公開されているサンプル・アプリケーション
https://github.com/oneapi-src/oneAPI-samples (英語)

© 2021 Intel Corporation.

法務上の注意書き

最適化に関する注意事項: インテル® コンパイラーでは、インテル® マイクロプロセッサーに限定されない最適化に関して、他社製マイクロプロセッサー用に同等の最適化を行えないことがあります。これには、インテル® ストリーミング SIMD 拡張命令 2、インテル® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサーに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサー依存の最適化は、インテル® マイクロプロセッサーでの使用を前提としています。インテル® マイクロアーキテクチャーに限定されない最適化のなかにも、インテル® マイクロプロセッサー用のものがあります。この注意事項で言及した命令セットの詳細については、該当する製品のユーザー・リファレンス・ガイドを参照してください。

性能に関するテストに使用されるソフトウェアとワークロードは、性能がインテル® マイクロプロセッサー用に最適化されていることがあります。SYSmark* や MobileMark* などの性能テストは、特定のコンピューター・システム、コンポーネント、ソフトウェア、操作、機能に基づいて行ったものです。結果はこれらの要因によって異なります。製品の購入を検討される場合は、他の製品と組み合わせた場合の本製品の性能など、ほかの情報や性能テストも参考にして、パフォーマンスを総合的に評価することをお勧めします。 さらに詳しい情報をお知りになりたい場合は、https://www.intel.com/benchmarks/ (英語) を参照してください。

ここに記載されているすべての情報は、予告なく変更されることがあります。

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。

結果は推定またはシミュレーションしたものです。

絶対的なセキュリティーを提供できる製品またはコンポーネントはありません。

インテルは、サードパーティーのデータについて管理や監査を行っていません。 ほかの情報も参考にして、正確かどうかを評価してください。

本資料は、(明示されているか否かにかかわらず、また禁反言によるとよらずにかかわらず) いかなる知的財産権のライセンスも許諾するものではありません。

インテルは、明示されているか否かにかかわらず、いかなる保証もいたしません。ここにいう保証には、商品適格性、特定目的への適合性、および非侵害性の黙示の保証、ならびに履行の過程、取引の過程、または取引での使用から生じるあらゆる保証を含みますが、これらに限定されるわけではありません。

© 2021 Intel Corporation. 無断での引用、転載を禁じます。

Intel、インテル、Intel ロゴ、VTune は、アメリカ合衆国および / またはその他の国における 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。

TECH.
DECODED

intel
Software