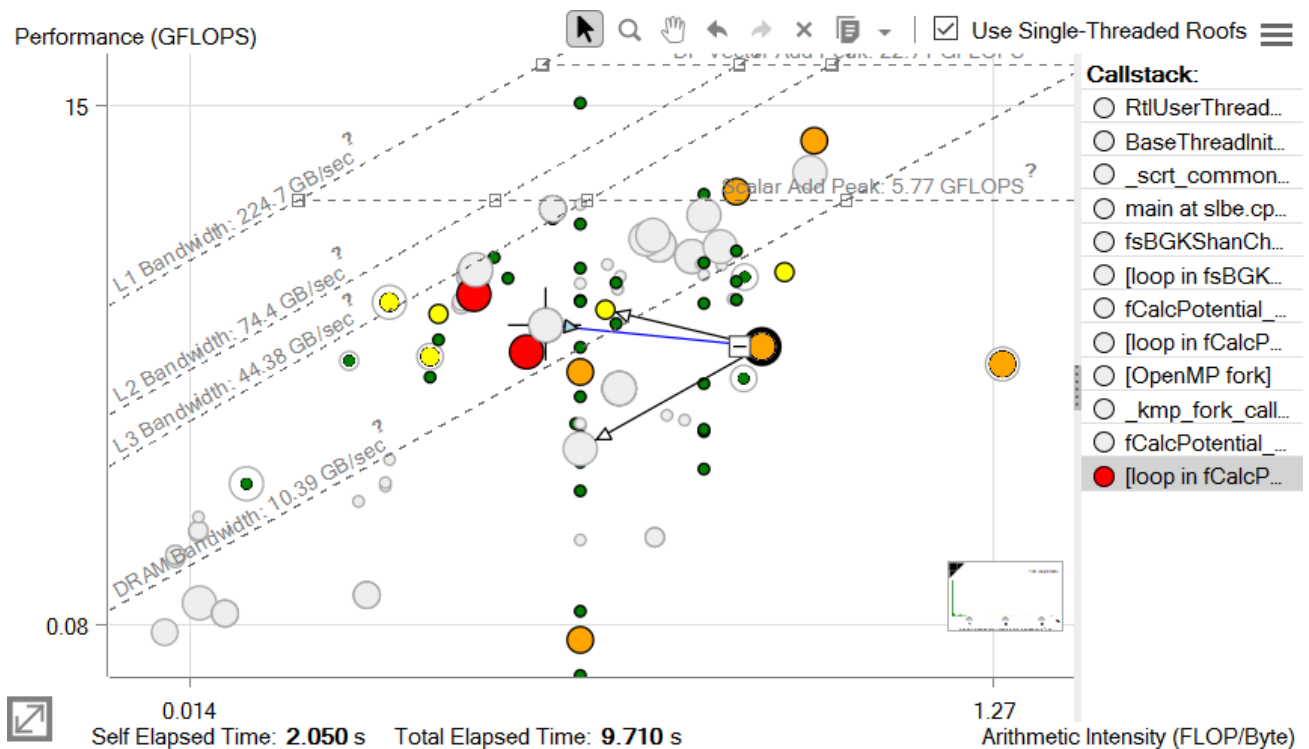


コールスタックを利用したルーフライン

[Alexandra S. \(Intel\)](#) 2017 年 12 月 1 日公開

この記事は、2017 年 12 月 18 日時点の、インテル® デベロッパー・ゾーンに公開されている「[Roofline with Callstacks](#)」の日本語訳です。



注: この記事の一部のスクリーンショットにはオレンジ色の点が表示されています。デフォルト設定では、これらの点は赤または黄色になります。赤と黄色の間のオレンジ色の点は、カスタマイズ・メニューで追加されたものです。

コールスタックを利用したルーフラインは、インテル® Advisor の[キャッシュを考慮したルーフライン](#)の拡張機能です。この機能は、2018 Update 1 で正式にサポートされました。2018 初期リリースでは、階層ルーフラインという名前のプレビュー機能でしたが、似た名前の異なる機能との混同を避けるため、現在の名前に変更されました。

オリジナルのキャッシュを考慮したルーフラインとコールスタックを利用したルーフラインの違いは、自己データと合計データの解釈です。自己データとは、呼び出されたほかの関数やループに起因するデータを除いた、ループや関数自体にのみ関連するデータ (メモリアクセス、FLOP、期間) です。合計データは、外部の関数やループ自体に起因するデータに加えて、その内部で呼び出された関数やループからのデータを含みます。

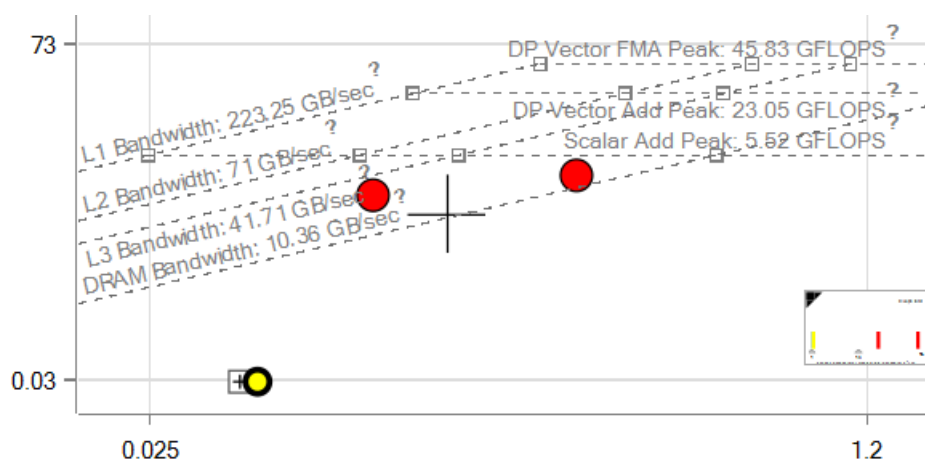
```

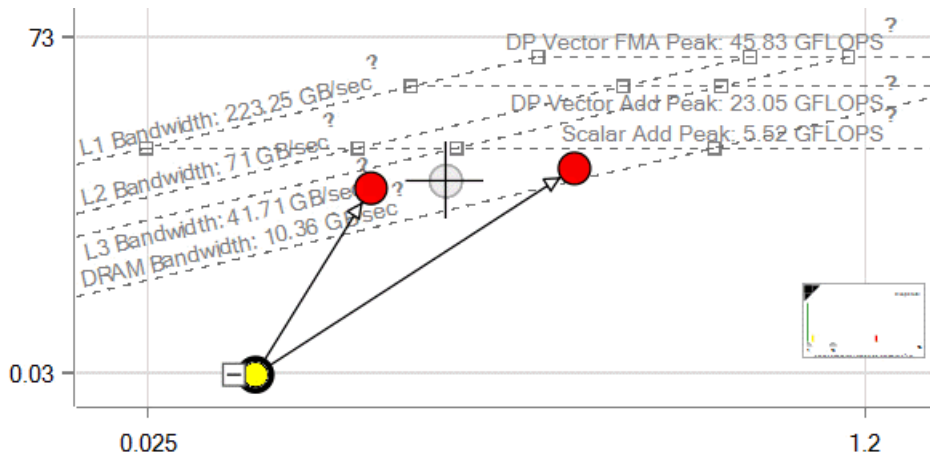
for (int x = 0; x < X_MAX; x++)
{
  /* これらの 3 の行は外部ループの自己データと
   * 合計データの両方にカウントされます */
  bar[x * 2] = x * 42.0 + 7.0;
  bar[(x * 2) - 1] = (x - 1.0) * 42.0 + 7.0;
  bar[x] = 23.9 * 83.9 / 31.2;
  /* foobar() と内部ループで実行される操作は
   * 外部ループの合計データにのみカウントされます */
  foobar();
  for (int y = 0; y < Y_MAX; y++)
  {
    foo[y*2] = y * 42.0 + 7.0;
  }
}

```

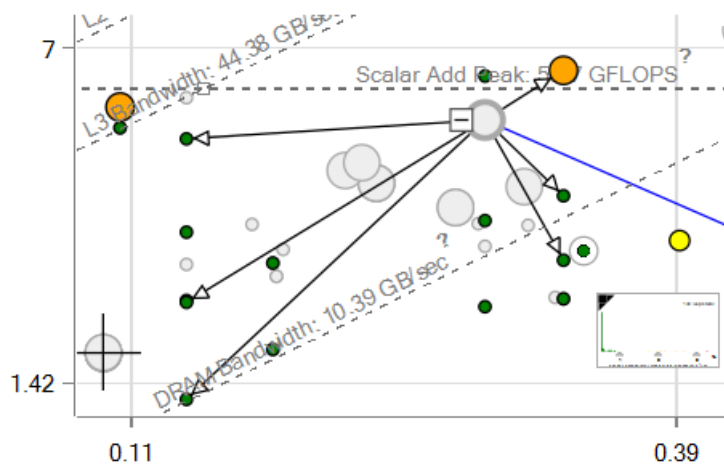
インテル® Advisor のオリジナルのキャッシュを考慮したルーフラインでは、自己データのみ表示され、点と点の関係は示されていませんでした。コールスタックを利用したルーフラインは、自己データに加えて合計データを利用します。また、関連する点を簡単に識別できるように、ナビゲーション可能なコールスタックとビジュアル・インジケータを提供します。

ユースケース





コールスタックを利用したルーファインは合計データを利用するため、データ粒度を調整できます。インテル® Advisor のオリジナルのルーファイン実装では、上記の例の外部ループの全体像を把握することは不可能でした。内部ループ、外部ループの自己データ、(何も呼び出していない) foobar() のみ表示されているためです。オリジナルのキャッシュを考慮したルーファインでは、最適化する価値のあるループに関する洞察は得られますが、同じ場所から呼び出される小さな点のクラスターに関する洞察は提供されません。



コールスタックを利用したルーファインは点の多いルーファインを単純化できます。いくつかの小さなループは開始ポイントを共有する場合があります。その場合、グラフ上の多くの小さな点を読むのではなく、点のクラスターの全体像に注目したほうが効果的です。

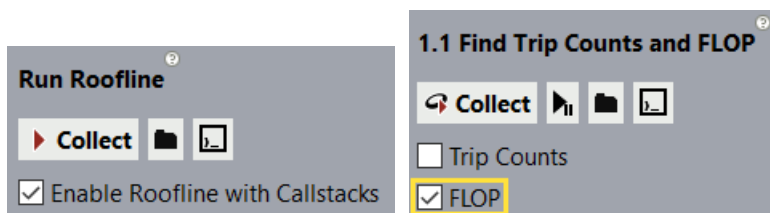
共有された開始ポイントを明らかにすることにより、開発者はループのみではなく、ループのソースを調査して、小さなループのパフォーマンスが低い根本的な原因となりうる非効率なコールチェーンを特定することができます。

コールスタックを利用したルーファインは、異なる状況の下で呼び出されたときに動作が異なる関数やループを正確に把握するのに非常に役立ちます。オリジナルのキャッシュを考慮したルーファインは関数ごとに 1 つの点を描画していましたが、呼び出し情報を含めることにより、コールスタックの異なる関数やループのインスタンスを区別して、それらの動作および特性を別々に解析することができます。

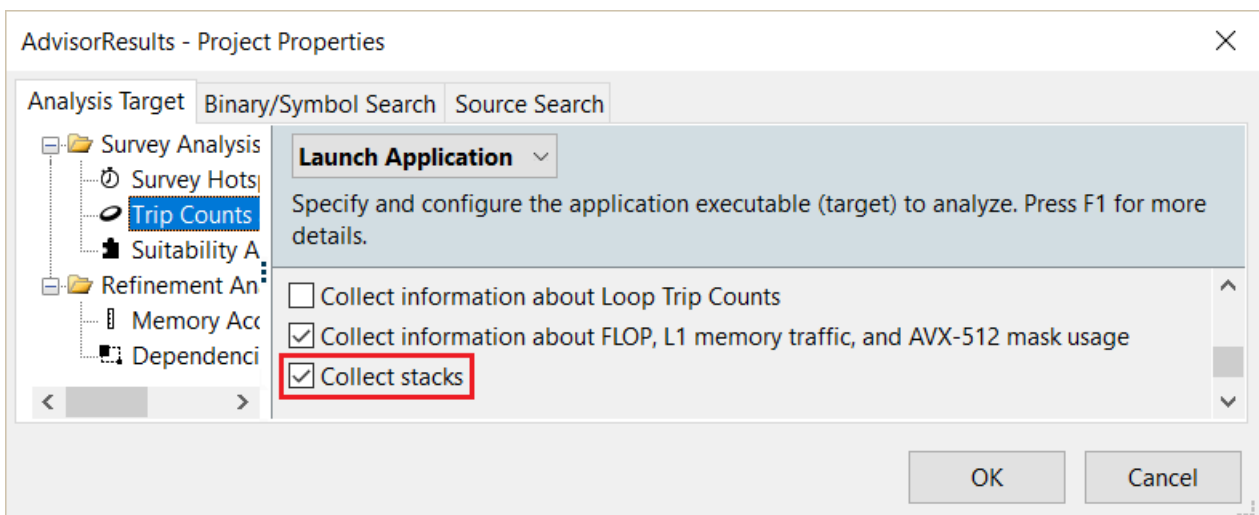
解析の実行

注: 2018 初期リリースを使用しているユーザーは、Intel® Advisor でコールスタックを利用したルーフラインのデータを収集または表示する前に、環境変数 `ADVIXE_EXPERIMENTAL=roofline_ex` を設定してプレビュー機能を有効にする必要があります。2018 Update 1 以降を使用しているユーザーは、この処理は必要ありません。

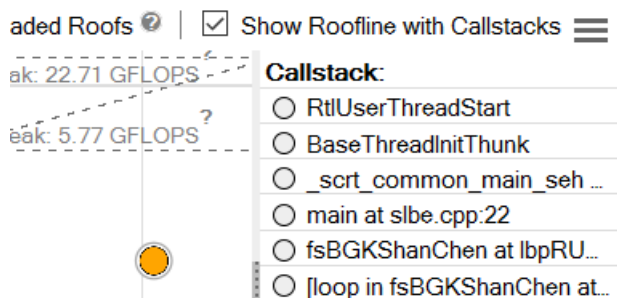
GUI



標準ルーフラインと同様に、[Run Roofline (ルーフライン実行)] ショートカット・ボタンを使用するか、サーベイ解析の後に FLOPS によるトリップカウント解析を実行して (トリップカウントはオプション)、ルーフライン解析を実行できます。どちらの場合でも、[Run Roofline (ルーフライン実行)] ショートカット・ボタンの下の [Enable Roofline with Callstacks (コールスタックを利用したルーフラインを有効にする)] チェックボックスをオンにする必要があります。



[Enable Roofline with Callstacks (コールスタックを利用したルーフラインを有効にする)] チェックボックスは、[Project Properties (プロジェクト・プロパティ)] の [Trip Counts and FLOPs Analysis (トリップカウントと FLOPS 解析)] タブの [Collect stacks (スタックを収集する)] チェックボックスと連動しています。一方のチェックボックスをオンにすると、もう一方のチェックボックスもオンになります。



結果を表示するときは、右上のルーフライン・オプションの横にある、[Show Roofline with Callstacks (コールスタックを利用したルーフラインを表示する)] チェックボックスがオンになっていることを確認してください。このチェックボックスが表示されていない場合は、チェックボックスが表示されるまでルーフラインの表示幅を広げてください。

コマンドライン

2018 Update 1 のコマンドラインでコールスタックを利用したルーフラインの情報を収集するには、次のコマンドを使用します。

```
advixe-cl -collect roofline -stacks -project-dir MyResults -- MyExecutable
```

サーベイ解析とトリップカウント解析を別々に収集する場合、またはすでに作業している標準サーベイレポートに FLOPS データを追加する場合は、コマンドラインで `-flop` オプションと `-stacks` オプションを使用します。

```
advixe-cl -collect survey -project-dir MyResults -- MyExecutable
```

```
advixe-cl -collect tripcounts -flop -stacks -project-dir MyResults -- MyExecutable
```

重要なのは FLOPS データであるため、トリップカウント・データの収集はオプションです。

注: ルーフライン収集タイプと上記で指定したオプションはどちらも 2018 初期リリースには存在しません。2018 初期リリースでは、環境変数が設定されていることを確認した後、次のオプションを指定して、標準のサーベイ収集およびトリップカウント収集を実行してください。

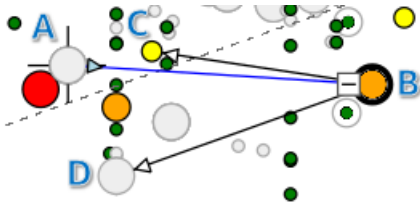
```
advixe-cl -collect survey -project-dir MyResults -- MyExecutable
```

```
advixe-cl -collect tripcounts -flops-and-masks -callstack-flops -project-dir  
MyResults - MyExecutable
```

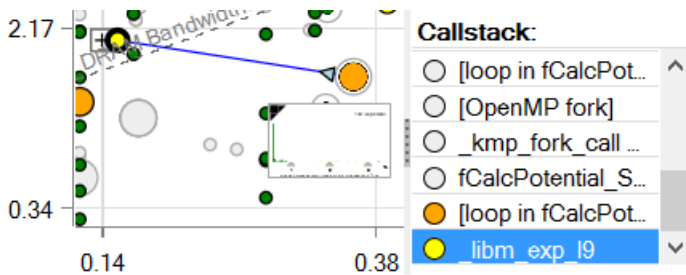
これらのオプションは Update 1 で非推奨になりました。`-flop` オプションおよび `-stacks` オプションを使用してください。

グラフを読む

コールスタックを利用したループラインは、ループライン・グラフにシンボルとビジュアル・インジケータを追加します。ループラインの基本的な読み方に関する情報は、[インテル® Advisor のループラインの記事](#)を参照してください。



新機能の 1 つは、呼び出し元/呼び出し先矢印です。グラフの点を選択すると、その直接の呼び出し元と呼び出し先が矢印で示されます。上の図では、オレンジ色の点 B が選択され、灰色の点 A が呼び出し元であることを示す逆矢印の青の線と、選択したループが黄色のループ C と灰色のループ D を直接呼び出していることを示す通常矢印の黒の線が示されています。



もう 1 つの新機能は、図の右のコールスタック表示です。このリストは、選択したループのコールチェーン全体を表示します (呼び出し先は除きます)。スタックの各レベルの横の点に、グラフの点の現在の色が表示されます。

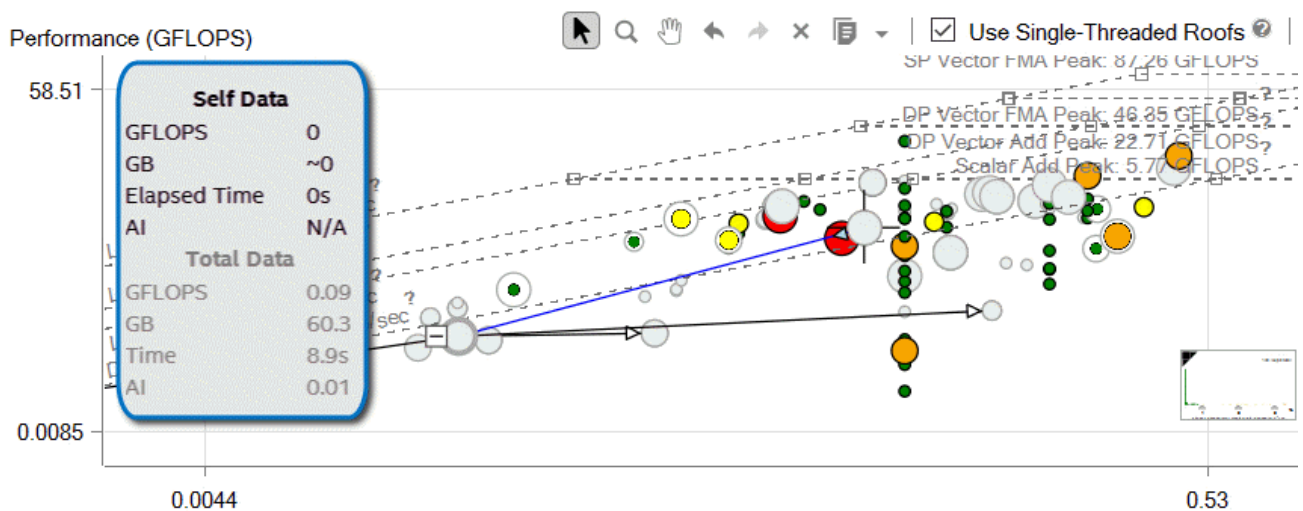
このリストの項目をクリックすると、ループライン・グラフの対応する点が点滅します。

Source	Top Down	Code Analytics
Function Call Sites and Loops		
[-] fsBGKShanChen\$omp\$parallel@199		
[-] [loop in fsBGKShanChen\$omp\$par		
[-] [loop in fsBGKShanChen\$omp\$		
[-] fCalcInteraction_ShanChen		
[-] [loop in fCalcInteraction_S		
[-] [loop in fCalcInteraction_S		
[-] [loop in fCalcInteraction		
[-] _chkstk		
[-] [loop in fCalcInteraction_S		

詳細なコールツリーは、[Top Down (トップダウン)] タブの下のペインに表示されます。1 つのコールチェーンではなく、ツリー全体が表示されます。分岐は、展開および折りたたむことができ、直接接続されていないノードを見つけるのに役立ちます。

同様に、ルーフライン・グラフの点も、プラスボタンとマイナスボタンを使用して展開する/折りたたむことができます。コールツリーと同様に、指定したノードを折りたたむと、そのノードの下の項目はすべて、直接の呼び出し先でないものも含め非表示になります。

点を折りたたむ/展開すると、点の表示のベースとなるデータが変更されます。自己データのないループおよび関数は、展開されたときに灰色で表示され、折りたたまれたときに色付きで表示されます。自己データのあるノードは、展開したときに適切な座標、サイズ、色で表示されますが、合計時間に対応したサイズの灰色の輪が付きます。これらのループを折りたたむと、合計時間に対応するサイズと色に変更され、必要に応じて、合計パフォーマンスと合計演算密度を反映するように移動します。



上の図の左に表示されている灰色の情報ボックスはインテル® Advisor では表示されません。分かりやすいように図に追加したものです。このボックスの情報は、インテル® Advisor の [Top Down (トップダウン)] タブの横にある [Code Analytics (コード解析)] タブに表示されます。[Code Analytics (コード解析)] タブは、いくつかの折りたたみセクションに、さまざまな有用なメトリックを含んでいます。自己演算密度と合計演算密度、メモリアクセス、経過時間、FLOPS に関する情報は、FLOPS 折りたたみセクションに表示されます。

最後に、ルーフライン・グラフの十字アイコンはアプリケーション全体を表し、プログラム全体の合計演算密度と合計 GFLOPS の座標に配置されることに注意してください。グラフのルートノードは、そのコールツリーにアプリケーション全体を含み、合計データはプログラム全体のデータになります。つまり、ルートノードは常に十字の場所に折りたたまれます。

コンパイラーの最適化に関する詳細は、[最適化に関する注意事項](#)を参照してください。

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

© 2018 Intel Corporation.