インテル® Advisor 2018 日本語版を使用して解析を始めましょう！

2018 年 9 月
iSUS 編集長
すがわら　きよふみ

内容

- 解析を始める前に
- インテル® Advisor ができること
- 操作と表示情報

© 2018 iSUS *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。

解析を始める前に: 利用状況の確認

1. 通常のアプリケーション開発 (ソース/シンボルあり)
2. MPI アプリケーション
3. すでに起動されているアプリケーション
4. プログラム (スクリプトなど) から起動されるアプリケーション
5. シンボルなしやほかのコンパイラーで生成されたバイナリー
6. ほかの OS で収集されたアプリケーションのデータ

ツールの起動方法:

- スタンドアロン GUI
- コマンドライン CLI
- Visual Studio* 統合環境から

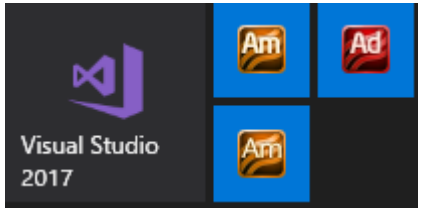
解析タイプ:

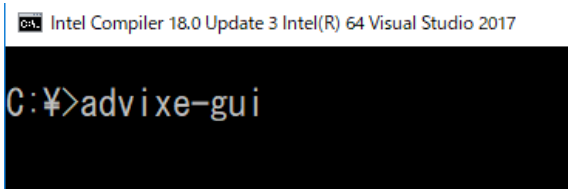
- サーベイ・ホットスポット
- トリップカウントと FLOP
- スータビリティーなど

© 2018 iSUS *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。

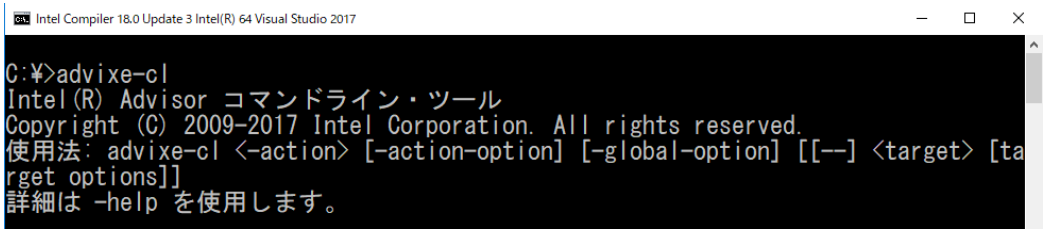
ツールの起動方法

スタンドアロンを起動

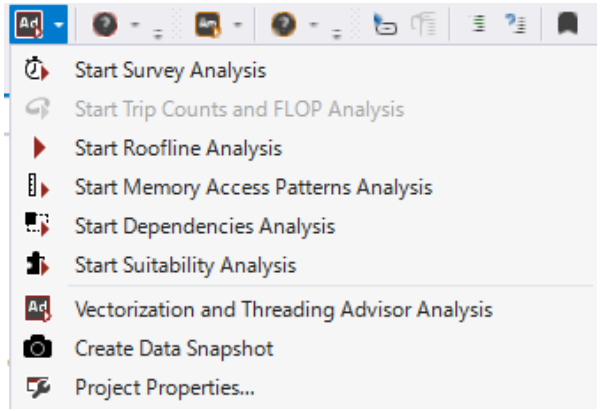




コマンドライン・ツールの起動

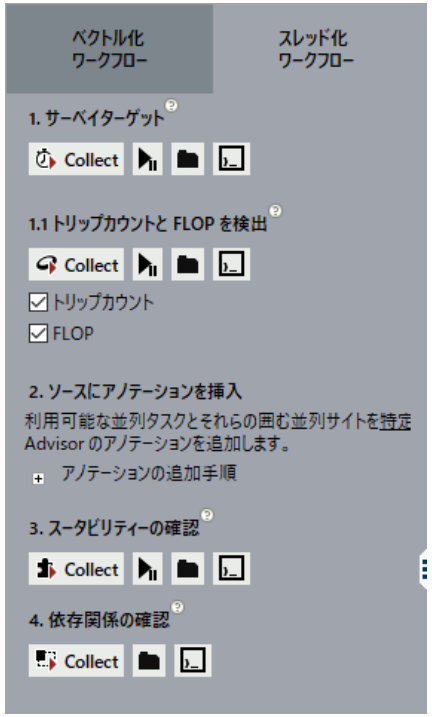


Visual Studio* から起動

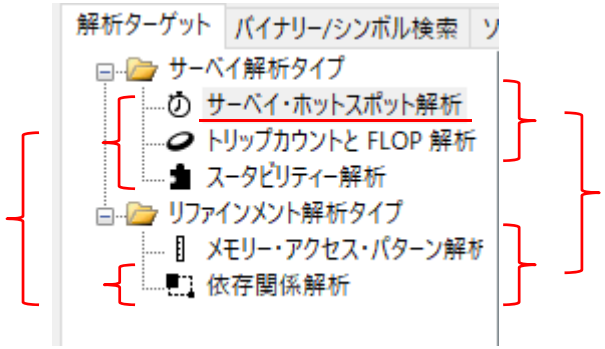


© 2018 iSUS *その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

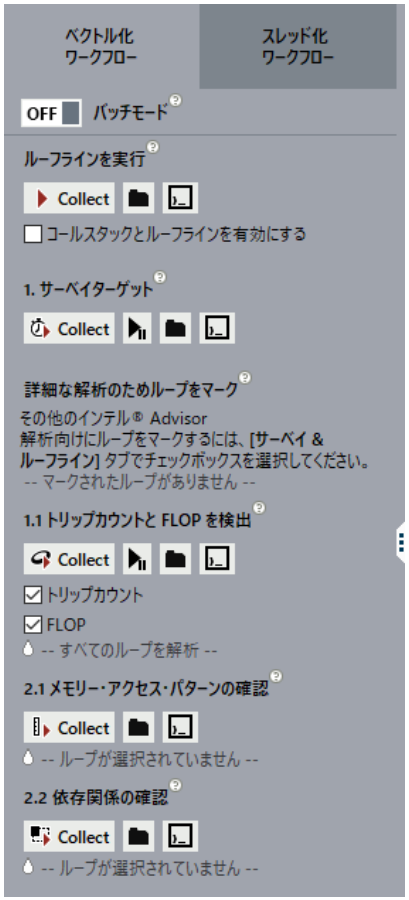
解析タイプの選択



プロジェクトのプロパティー



基本は、「サーベイ・ホットスポット解析」から始めます

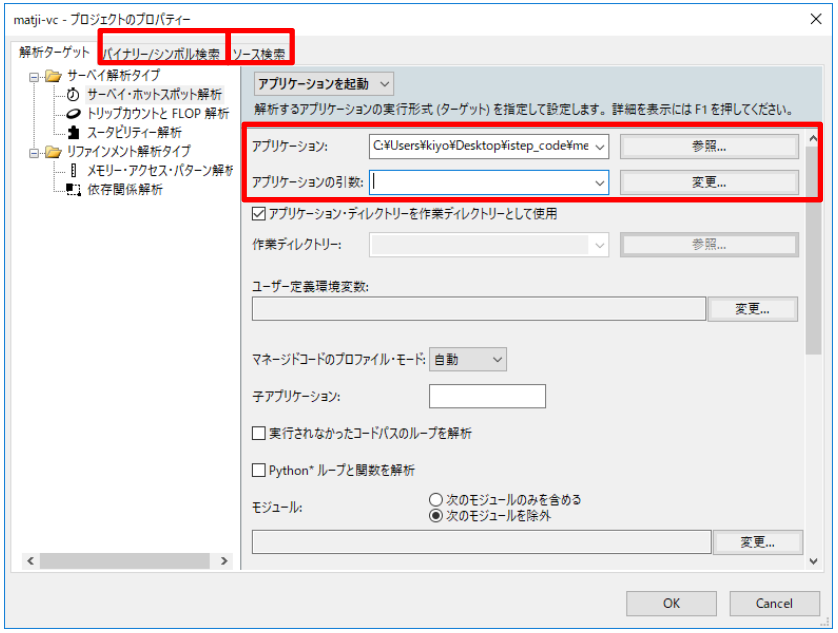


© 2018 iSUS *その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

解析を始める前に 1:
ソースとバイナリーの場所を確認します

[プロジェクトのプロパティー] で、
- アプリケーション、
- バイナリー/シンボル検索、
- ソース検索

を設定します



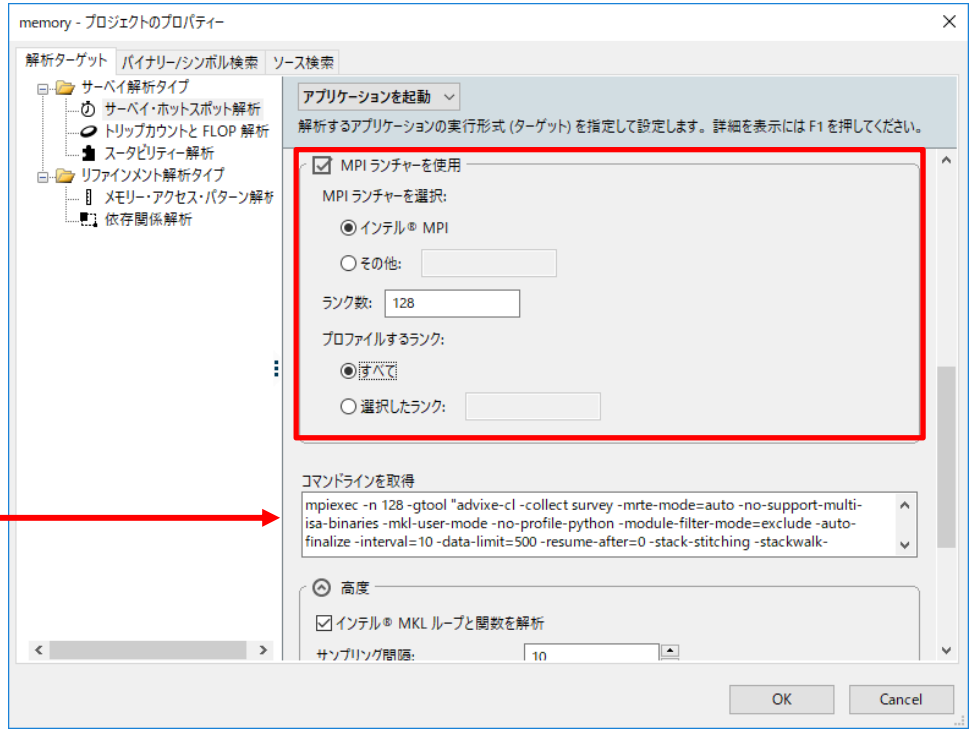
起動する子アプリケーションを解析する場合は、[子アプリケーション] を指定します

© 2018 iSUS *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。

解析を始める前に 2:
MPI ランチャーを使用する場合

MPI 環境で解析を行う場合、[アプリケーション] を指定して、MPI ランチャーに関連する情報を指定します

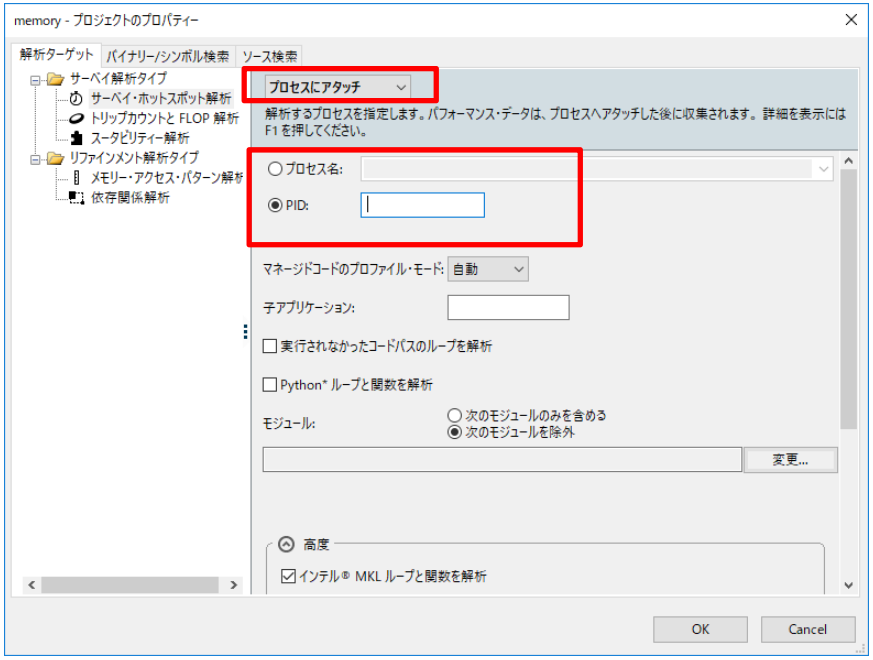
ノード上で実行するコマンドラインが生成されます



© 2018 iSUS *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。

解析を始める前に 3:
プロセスにアタッチする場合は PID を確認

[プロジェクトのプロパティー] で、[プロセスにアタッチ] を選択して、プロセス名または PID を設定します

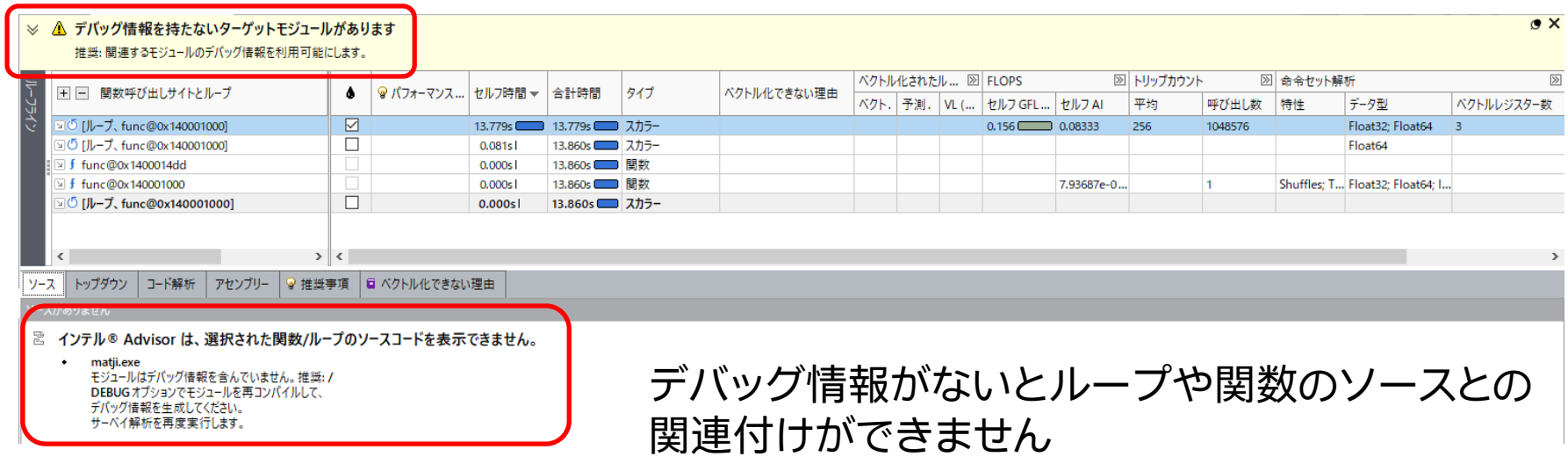


起動する子アプリケーションを解析する場合は、[子アプリケーション] を指定します

© 2018 iSUS *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。

解析を始める前に 4: デバッグ情報を生成します

インテル® Advisor 2018 を使用してアプリケーションの解析を行う前に、プログラムのデバッグ情報を生成します



デバッグ情報がないとループや関数のソースとの関連付けができません

© 2018 iSUS *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。

解析を始める前に 5:
ほかのコンパイラーで生成されたバイナリーの解析

- インテル® Advisor 2018 は、Visual C++*、GNU* gcc、およびその他の C/Fortran コンパイラーと互換性があります

  ソースコードと関連付けて解析を行うためシンボル情報を生成します: /Zi、/Zl など (Windows*)、-g (Linux*)

インテル® C++/Fortran コンパイラー以外で生成されたバイナリーは、シンボル情報を持っていても一部の解析情報が表示されないことがあります

© 2018 iSUS *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。

ほかのコンパイラーで生成されたコード

関数呼び出しサイトとループ		パフォーマンス問題	セルフ時間	合計時間	タイプ	ベクトル化できない...	ベクトル化されたル...			FLOPS		トリップカウント		命令セット解析			詳細	場所
							ベクト.	予測.	VL (...	セルフ GFL...	セルフ AI	平均	呼び出し数	特性	データ型	ベクト.		
[ループ、mem.c:34 の main]	☑		0.523s	0.523s	ベクトル...		SSE2		2; 4	4.107	0.08333	128	1048576		Float32...	3		mem.c:34
_scrt_common_main_seh	☐		0.000s	0.523s	関数								1			0		exe_common.inl:231
main	☐		0.000s	0.523s	関数						7.93686e-0...		1	Shuffles; Type Convers...	Float32...	4		mem.c:14
[ループ、mem.c:32 の main]	☐		0.000s	0.523s	スカラー							1024	1024	Unpacks	Float64...	3		mem.c:32
[ループ、mem.c:31 の main]	☐		**0.000s**	**0.523s**	**スカラー**							**1024**	**1**			**0**		**mem.c:31**

インテル® コンパイラー以外で生成されたバイナリーもデバッグ情報が提供されていれば、ルーフライン、サーベイ、トリップカウントと FLOP、メモリー・アクセス・パターン、および依存関係解析の大部分の機能を利用できます

利用できない機能もあります

© 2018 iSUS *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。

解析を始める前に 6:
ほかの OS で収集されたデータの解析

インテル® Advisor 2018 for Windows* は、Linux* 上で収集されたデータを解析できます。Windows* 上で解析する前に次の操作を行います:

- 収集結果をローカルにコピーします
- ソースとバイナリーをローカルにコピーします
- インテル® Advisor 2018 for Windows* を起動後、プロジェクトの再設定を行います

詳細は、こちらから

© 2018 iSUS *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。

解析を始める前に 7:
コンパイルオプションを確認

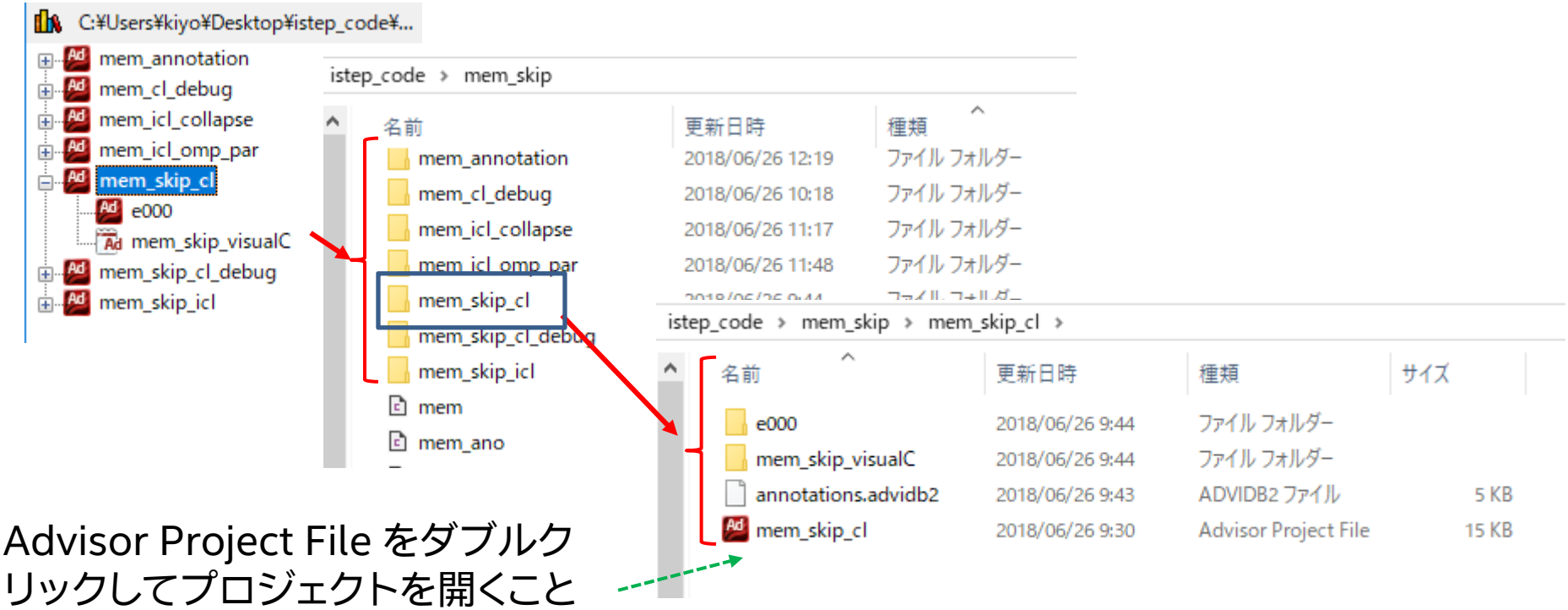
インテル® Advisor 2018 から提示される解析結果や推奨事項を理解できるよう、バイナリーのコンパイルに使用したコンパイルオプションを確認しておきましょう

インテル® コンパイラーを使用する場合、レポート機能 (/Qopt-report) を使用すると、生成されるレポート内にオプションが示されます:

コンパイルオプションの例:
/O1 /QaxCORE-AVX512 /Qopt-zmm-usage:high /Zi /Qopt-report3 /Qopt-report-phase:vec

© 2018 iSUS *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。

プロジェクトと解析データの構成



Advisor Project File をダブルクリックしてプロジェクトを開くことができます

© 2018 iSUS *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。

update1、2、3 の機能

update3:

- 整数 OP/秒ベースのルーフライン – 整数データ型と浮動小数点/整数データ型が混在するコード向けのルーフライン・グラフ
- アプリケーションの実用パフォーマンスの限界を確認するため、カスタムスレッド数にルーフラインを調整する機能

update2:

- プロジェクトのプロパティーでループ呼び出し数を制限し、解析時間を設定することで、メモリー・アクセス・パターンと依存関係解析のオーバーヘッドを軽減

update1:

- 解析範囲を狭め、オーバーヘッドを軽減するため、ルーフライン、FLOPS とトリップカウント収集でプロファイルを選択する機能を追加

https://www.isus.jp/products/advisor/advisor-2018-update-3-release-notes-whats-new/

© 2018 iSUS *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。

内容

- 解析を始める前に
- インテル® Advisor ができること
- 操作と表示情報

© 2018 iSUS *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。

インテル® Advisor ができること

1. サーベイ解析
2. メモリー･アクセス･パターンの解析
3. 依存関係の解析
4. ルーフライン解析
5. スータビリティー解析

© 2018 iSUS *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。

インテル® Advisor ができること

1. サーベイ解析
   - アプリケーションに含まれる命令セット
     - 命令セット/効率/予測されるゲイン/データ型
   - 関数とループにかかる時間
     - セルフ時間と合計時間
   - ループがどのように最適化/ベクトル化されているか
     - ベクトル化/ループ変換/アンロール
   - 関数がどのように最適化 (インライン展開) されているか

© 2018 iSUS *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。

インテル® Advisor ができること

2. メモリー・アクセス・パターンの解析
   - 重要なベクトルコードの実行速度を低下させたり、コンパイラーによる自動ベクトル化の妨げとなる問題を排除するため、メモリーアクセスを解析します

   - 不連続なメモリーアクセスや、ユニットストライド対非ユニットストライドなアクセスなど、多様なメモリーアクセスの問題を調査します

© 2018 iSUS *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。

インテル® Advisor ができること

3. 依存関係の解析
   - コンパイラーは安全性のため、データに依存関係が想定されるとしばしば保守的なコード生成を行います
   - 依存関係が想定されたため、コンパイラーがベクトル化できなかったループの実際の依存性を確認するには、依存関係解析を実行します
   - 実際に依存関係が検出されると、解析は依存関係を解決するのに役立つ詳細な情報を提供します

目的: ベクトル化の強制が安全ではない真のデータ依存関係を特定して分類します

© 2018 iSUS *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。

インテル® Advisor ができること

4. ルーフライン解析
   - ハードウェアによって課せられるパフォーマンスの上限に対して、制限の主な要因 (メモリー帯域幅や計算能力) を検証して、実際のパフォーマンスを可視化するのを支援します。これにより、潜在的な最適化ステップの理想的なロードマップが示されます

次のような疑問に答えるため**ルーフライン**・チャートが使用されます:
- 現在のハードウェア・リソースで達成可能なパフォーマンスの上限は？
- 現在のハードウェア・リソースでアプリケーションが適切に動作するか?
- そうでない場合、最初の最適化の候補は何か？
- メモリー帯域幅や計算能力は、各最適化候補のパフォーマンスを制限しているか

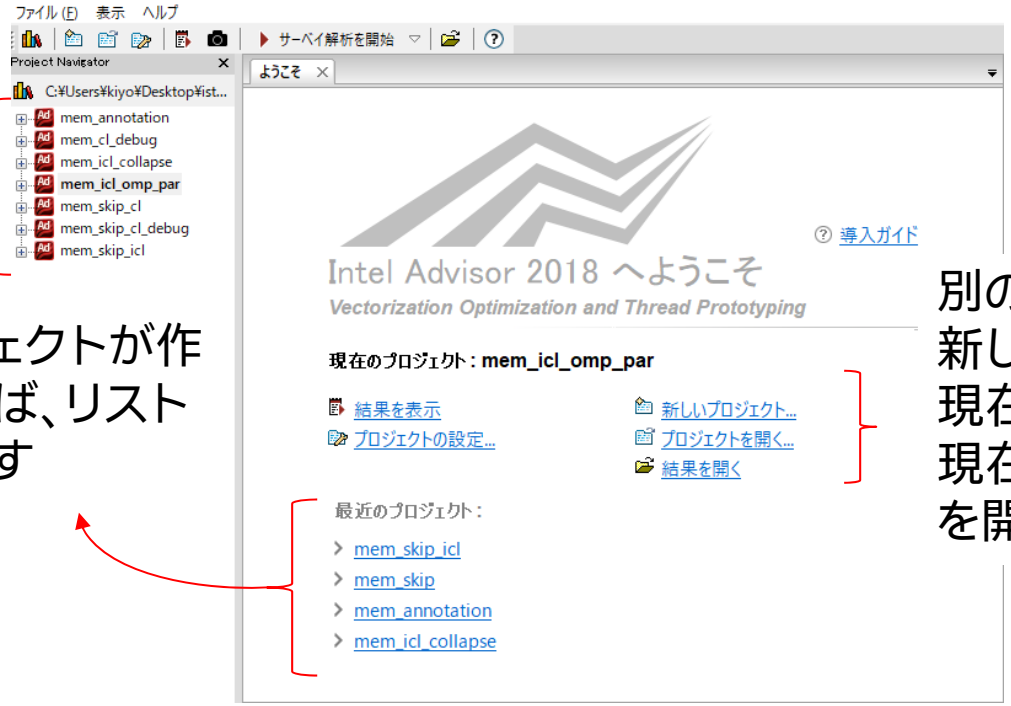
© 2018 iSUS *その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

内容

- 解析を始める前に
- インテル® Advisor ができること
- 操作と表示情報

© 2018 iSUS *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。

ようこそ画面: プロジェクトの作成や制御



すでにプロジェクトが作成されていれば、リストに表示されます

別のプロジェクトを開く、新しいプロジェクトの作成、現在のプロジェクト設定、現在のプロジェクトの結果を開く、など制御できます

Visual Studio* から起動される場合、この画面は表示されません

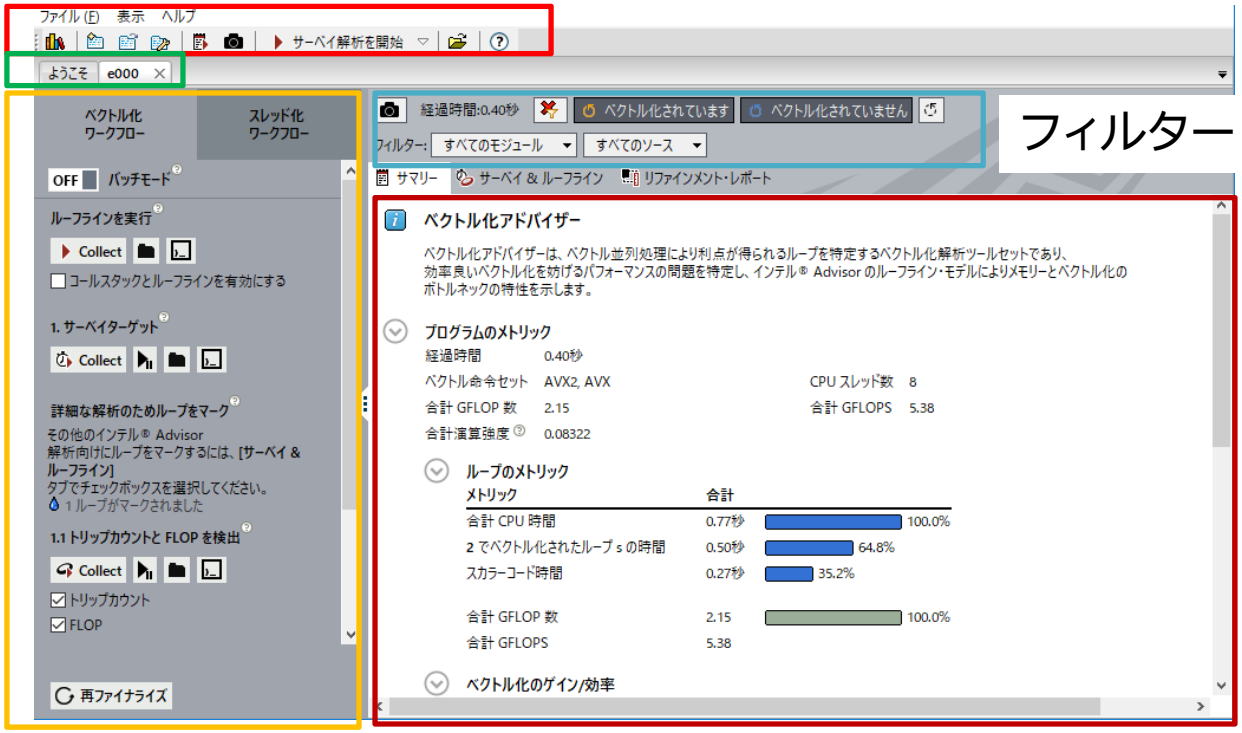
© 2018 iSUS *その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

インテル® Advisor 操作画面の構成

メニューとツールバー

結果タブ

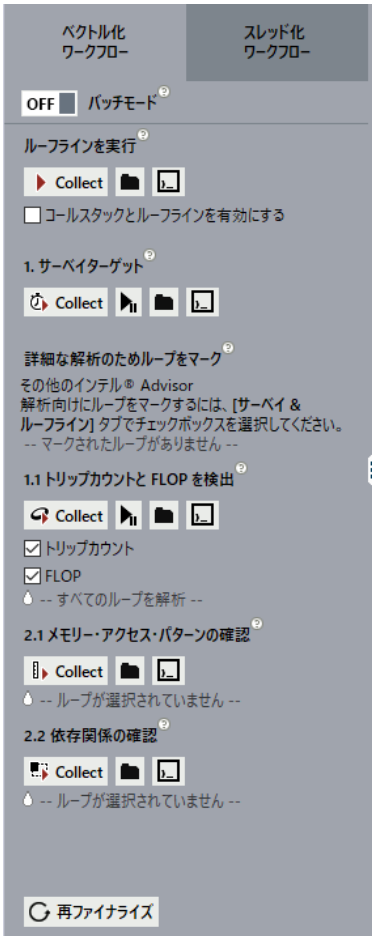
ワークフロー



フィルター

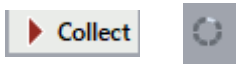
解析結果表示

© 2018 iSUS *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。



ワークフロー・タブ

ベクトル化とスレッド化のワークフローを切り替えます

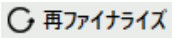


クリックしてデータの収集を開始します

収集を停止した状態でコレクターを起動します
収集は API で制御します

レポートの結果を開きます

コマンドラインを取得します



データを再度ファイナライズします

© 2018 iSUS *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。

ベクトル化アドバイザー

© 2018 iSUS *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。

解析結果: サーベイ解析

上段にサーベイ結果、下段に対応するソース情報が表示されます

関数呼び出しサイトとループ	パフォーマンス...	セルフ時間	合計時間	タイプ	ベクトル化できない理由	ベクトル化されたループ				
						ベクトル ISA	効率	予測ゲイン	VL (ベクトル長)	コンパイラーの予測...
[ループ、mem.c:41 の mainompparallel_for@		0.497s	0.497s	ベクトル化 (本体)		AVX2	100%	5.15x	4	5.14x
[ループ、mem.c:40 の mainompparallel_for@3		0.010s	0.507s	スカラー	loop was not vectorized: inne...					
[ループ、mem.c:21 の main]		0.000s	0.000s	ベクトル化 (本体)		AVX	100%	4.89x	4	4.89x
[ループ、mem.c:39 の mainompparallel_for@3		0.000s	0.507s	スレッド化 (Open...	loop was not vectorized: inne...					
[ループ、mem.c:20 の main]	1 データ型の...	0.000s	0.000s	スカラー	loop was not vectorized: inne...					

FLOPS						トリップカウント		命令セット解析			詳細	場所
セルフ GFLOPS	セルフ AI	セルフ GFLOP	セルフメモリー (GB)	セルフ経過時間	合計経過時間	平均	呼び出し数	特性	データ型	ベクトルレジスター数		
27.488	0.08333	2.147	25.770	0.078秒	0.078秒	256	1048576	FMA	Float64	2		mem.c:41
		0	0.008	0.010秒	0.078秒	1024	1024		Float64	1		mem.c:40
		0	0.025	< 0.001秒	< 0.001秒	64	1024		Float64	3	アンロール by 4	mem.c:21
				0秒	0.078秒	128	8			0		mem.c:39
		0	< 0.001	0秒	< 0.001秒	1024	1	Type Conversions	Float64...	1		mem.c:20

上段

ソース | トップダウン | コード解析 | アセンブリー | 推奨事項 | ベクトル化できない理由

ファイル: mem.c:20 main

行	Source	合計時間	%	ループ/関数時間	%	特性
20	for(i=0; i<n; i++){			54.911µs		
	[ループ、mem.c:20 の main] スカラー ループ AVX2 レジスターを使用する命令を含む. Loop was not vectorized: inner loop was already vec ループ変換は適用されていません					
21	for(j=0; j<m; j++) {					
22	a[i][j] = 1.0;					
23	b[i][j] = i+1;					Type Conversions
24	c[i][j] = 0.0;					

下段

© 2018 iSUS *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。

解析結果: サーベイ解析のトップダウン

サマリー | サーベイ & ルーフライン | リファインメント・レポート

INTEL ADVISOR 2018

ルーフライン

関数呼び出しサイトとループ	パフォーマンス...	セルフ時間	合計時間	タイプ	ベクトル化できない理由	ベクトル化されたループ					
						ベクトル ISA	効率	予測ゲイン	VL (ベクトル長)	コンパイラーの予測...	
[ループ、mem.c:41 の mainompparallel_for@		0.497s	0.497s	ベクトル化 (本体)		AVX2	100%	5.15x	4	5.14x	2
[ループ、mem.c:40 の mainompparallel_for@3		0.010s	0.507s	スカラー	loop was not vectorized: inne...						
[ループ、mem.c:21 の main]		0.000s	0.000s	ベクトル化 (本体)		AVX	100%	4.89x	4	4.89x	
[ループ、mem.c:39 の mainompparallel_for@3		0.000s	0.507s	スレッド化 (Open...	loop was not vectorized: inne...						
[ループ、mem.c:20 の main]	1 データ型の...	0.000s	0.000s	スカラー	loop was not vectorized: inne...						

ソース | トップダウン | コード解析 | アセンブリー | 推奨事項 | ベクトル化できない理由

関数呼び出しサイトとループ	合計時間 %	合計時間	セルフ時間	タイプ	ベクトル化できない理由	ベクトル化されたル...			命令セット解析		詳細	場所
						ベクト.	VL (...	コンパ.	特性	データ型		
合計	100.0%	0.767秒	0.000s									
RtlUserThreadStart	100.0%	0.767秒	0.000s	関数								
BaseThreadInitThunk	100.0%	0.767秒	0.000s	関数								
_scrt_common_main_seh	82.8%	0.635秒	0.000s	関数								exe_common...
main	82.8%	0.635秒	0.000s	関数					FMA	Float64		mem.c:14
[OpenMP fork]	66.1%	0.507秒	0.000s	関数								kmp_csuppor...
_kmp_fork_call	66.1%	0.507秒	0.000s	関数					Shifts; Shuffles; Unpacks	Float32...		kmp_runtime....
[OpenMP dispatcher]	66.1%	0.507秒	0.000s	関数								kmp_runtime....
mainompparallel_for@3	66.1%	0.507秒	0.000s	関数					FMA	Float64		mem.c:32
[ループ、mem.c:39 の	66.1%	0.507秒	0.000s	スレッド化 (OpenMP*)	loop was not vectoriz...							mem.c:39
[ループ、mem.c:40	66.1%	0.507秒	0.010s	スカラー	loop was not vectoriz...					Float64		mem.c:40
[ループ、mem.c:	64.8%	0.497秒	0.000s	ベクトル化 (本体)		AVX2	4	5.14x	FMA	**Float64**		mem.c:41
_kmp_join_call	0.0%	0秒	0.000s	関数								kmp_runtime....
_kmp_get_global_thread_id_reg	16.7%	0.128秒	0.128s	関数						Float32		kmp_runtime....
[ループ、mem.c:20 の main]	0.0%	< 0.001秒	0.000s	スカラー	loop was not vectoriz...				Type Conversions	Float64...		mem.c:20
[OpenMP worker]	17.2%	0.132秒	0.000s	関数						Float32		z_Windows_...

© 2018 iSUS *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。

解析結果: サーベイ解析のコード解析

© 2018 iSUS *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。

解析結果: サーベイ解析のアセンブリー

サマリー | サーベイ & ループライン | リファインメント・レポート

ループライン

関数呼び出しサイトとループ	パフォーマンス...	セルフ時間	合計時間	タイプ	ベクトル化できない理由	ベクトル化されたループ			
						ベクトル ISA	効率	予測ゲイン	V
[ループ、mem.c:41 の mainompparallel_for@		0.497s	0.497s	ベクトル化 (本体)		AVX2	100%	5.15x	4
[ループ、mem.c:40 の mainompparallel_for@3		0.010s	0.507s	スカラー	loop was not vectorized: inne...				
[ループ、mem.c:21 の main]		0.000s	0.000s	ベクトル化 (本体)		AVX	100%	4.89x	4
[ループ、mem.c:39 の mainompparallel_for@3		0.000s	0.507s	スレッド化 (Open...	loop was not vectorized: inne...				
[ループ、mem.c:20 の main]	1 データ型の...	0.000s	0.000s	スカラー	loop was not vectorized: inne...				

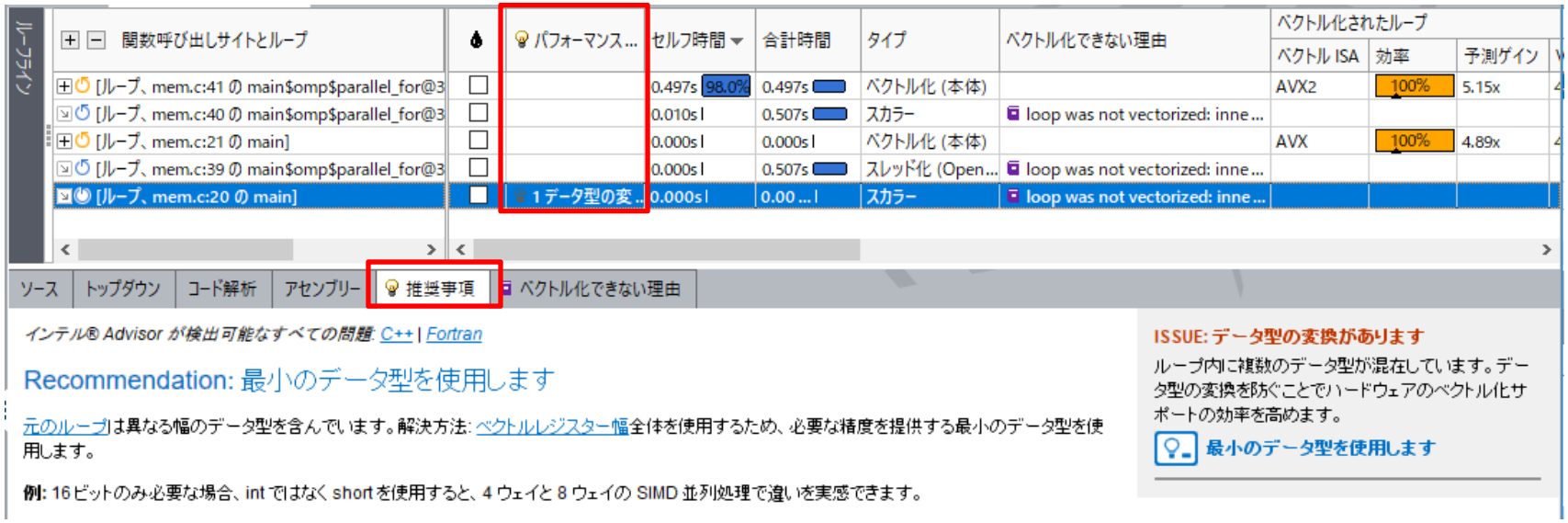
ソース | トップダウン | コード解析 | アセンブリー | 推奨事項 | ベクトル化できない理由

モジュール: mem_omp_par.exe!0x140006c38

アドレス	行	アセンブリー	合計時間	%	セルフ時間	%	特性
0x140006c38		**Block 1: 268435456**					
0x140006c38	42	lea r14, ptr [r11+rbx*8+0x883be0]					
0x140006c40	42	vmovupd ymm1, ymmword ptr [r14+rdi*1]	34.512ms		34.512ms		
0x140006c46	42	lea rdx, ptr [rcx+rbx*8]	200.761ms		200.761ms		
0x140006c4a	42	vfmadd213pd ymm1, ymm0, ymmword ptr [r11+rdx*1+0x1083be0]	50.032ms		50.032ms		FMA
0x140006c54	41	add rbx, 0x4	166.101ms		166.101ms		
0x140006c58	42	vmovupd ymmword ptr [r11+rdx*1+0x1083be0], ymm1	15.625ms		15.625ms		
0x140006c62	41	cmp rbx, rax	29.934ms		29.934ms		
0x140006c65	41	jb 0x140006c38 <Block 1>					
0x140006c7a		**Block 2:**					
0x140006c7a	42	lea rbx, ptr [r11+rsi*8+0x883be0]					
0x140006c82	42	vmovsd xmm1, qword ptr [rbx+rdi*1]					
0x140006c87	42	lea r14, ptr [rcx+rsi*8]					
0x140006c8b	42	vfmadd213sd xmm1, xmm0, qword ptr [r11+r14*1+0x1083be0]					FMA
0x140006c95	41	inc rsi					
0x140006c98	42	vmovsd qword ptr [r11+r14*1+0x1083be0], xmm1					
0x140006ca2	41	cmp rsi, r8					
0x140006ca5	41	jb 0x140006c7a <Block 2>					

© 2018 iSUS *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。

解析結果: サーベイ解析の推奨事項



© 2018 iSUS *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。

解析結果: サーベイ解析のベクトル化できない理由

サマリー | サーベイ & ルーフライン | リファインメント・レポート

ルーフライン

関数呼び出しサイトとループ	パフォーマンス...	セルフ時間	合計時間	タイプ	ベクトル化できない理由	ベクトル化されたループ: ベクトル ISA	効率	予測ゲイン
[ループ、mem.c:41 の mainompparallel_for@3		0.497s 98.0%	0.497s	ベクトル化 (本体)		AVX2	100%	5.15x
[ループ、mem.c:40 の mainompparallel_for@3		0.010s	0.507s	スカラー	loop was not vectorized: inne...			
[ループ、mem.c:21 の main]		0.000s	0.000s	ベクトル化 (本体)		AVX	100%	4.89x
[ループ、mem.c:39 の mainompparallel_for@3		0.000s	0.507s	スレッド化 (Ope...	loop was not vectorized: inne...			
[ループ、mem.c:20 の main]	1 データ型の変...	0.000s	0.00...	スカラー	loop was not vectorized: inne...			

ソース | トップダウン | コード解析 | アセンブリー | 推奨事項 | ベクトル化できない理由

すべてのコンパイラー診断

内部ループはすでにベクトル化されています

原因: 入れ子になったループの内部ループがベクトル化されています。

C++ の例:

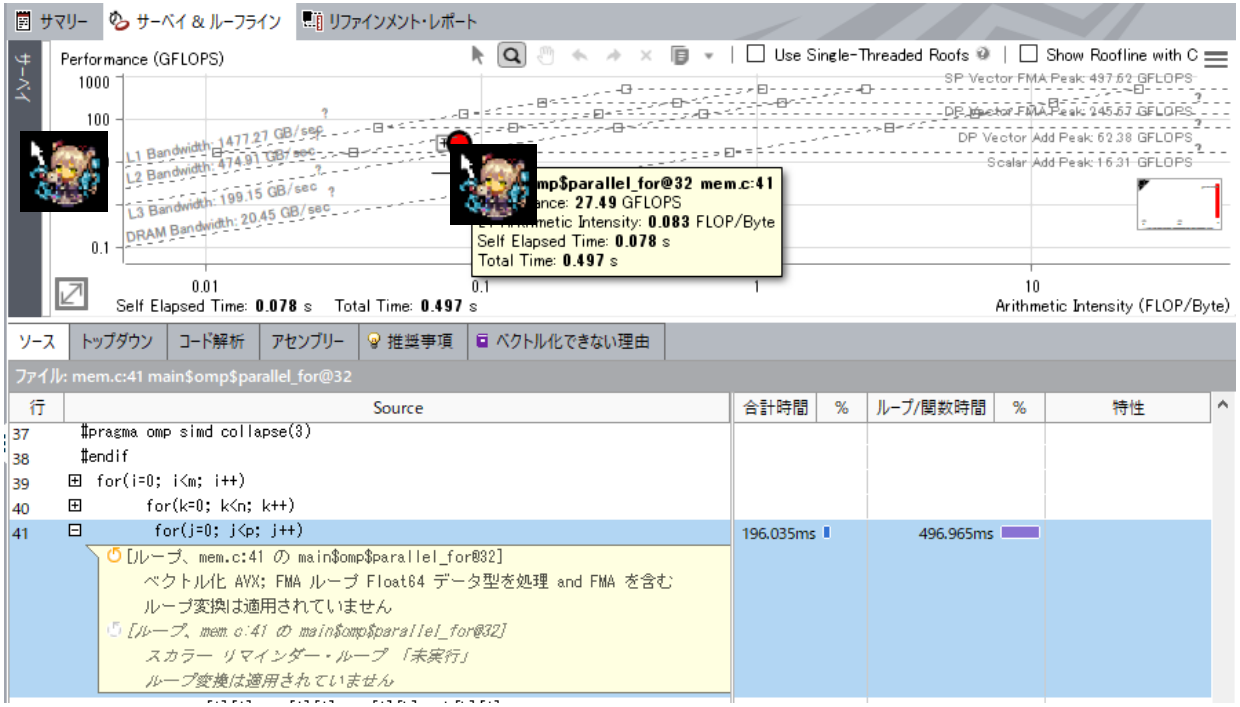
```
#define N 1000
float A[N][N];
void foo(int n) {
     int i,j;
     for (i = 0; i < n; i++) {
          for (j = 0; j < n; j++) {
               A[i][j]++;
          }
     }
}
```


© 2018 iSUS *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。

解析結果:ルーフライン解析

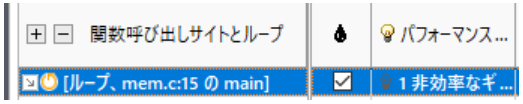
「サーベイ」の縦バーをクリックするとサーベイ表示に切り替えられます

ルーフラインの点(ドット) をクリックすると対応するソース行が下段に表示されます

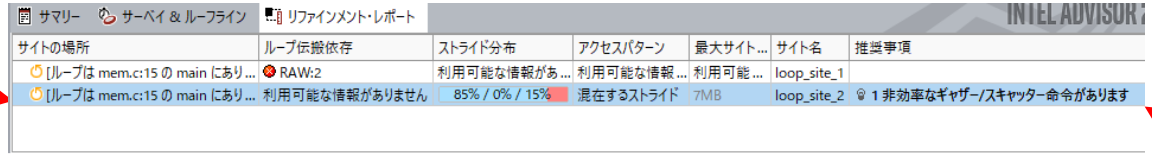


© 2018 iSUS *その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

解析結果: メモリー･アクセス･パターン解析



サーベイ結果で詳細解析を行うループを選択

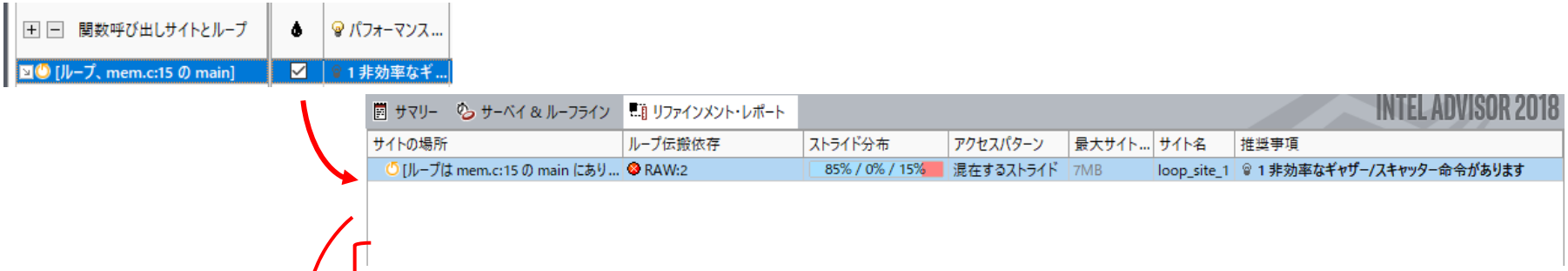


クリックして説明を表示

リファインメント･レポートの行をクリックしてメモリー･アクセス･パターン情報を表示します

© 2018 iSUS *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。

解析結果: 依存関係解析



リファインメント・レポートの行をクリックしてメモリー・アクセス・パターン情報を表示します

© 2018 iSUS *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。

スレッド化アドバイザー

© 2018 iSUS *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。

解析結果: スータビリティー (適合性) 解析

アプリケーションに並列化フレームワークを追加して変更する前に、アノテーションを使用してシリアルプログラムのパフォーマンスと依存関係の問題を推測します

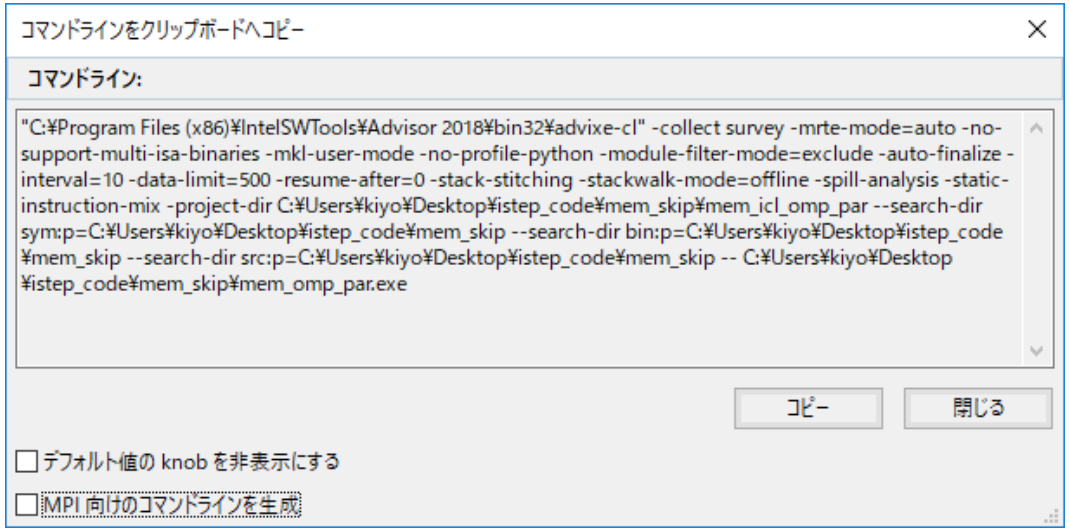
© 2018 iSUS *その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

その他の機能

- コマンドライン･ツール
- 環境変数
- 特殊アノテーション

© 2018 iSUS *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。

コマンドライン・ツールを使用



各種解析で、「コマンドラインを取得」機能が用意されています

クリップボードへコピーして、コマンドラインから直接実行できます

コマンドラインの詳細については、ヘルプの「コマンドライン・インターフェイスのリファレンス」を参照してください

© 2018 iSUS *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。

環境変数

- 環境変数 ADVISOR_201n_DIR を使用してコンパイル時に必要な追加のインクルード･ディレクトリーを指定することができます
- 評価する機能が有効化されているかどうかを調べるには、ADVIXE_EXPERIMENTAL 環境変数をテストします

事前公開機能など、一部の機能を利用するには環境変数を使用して有効化する必要があります:

ADVIXE_EXPERIMENTAL=int_roofline (update3 で追加)
ADVIXE_EXPERIMENTAL=roofline_ex (update2 で追加)
ADVIXE_EXPERIMENTAL=cachesim (update1 で追加)

© 2018 iSUS *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。

インテル® Advisor 特殊アノテーション API

Pause 収集と Resume 収集アノテーションを使用することで、データ収集をポーズおよび再開して、プログラム実行の不要な部分をスキップすることができます。

収集のポーズ:

C/C++: ANNOTATE_DISABLE_COLLECTION_PUSH;

Fortran: call annotate_disable_collection_push()

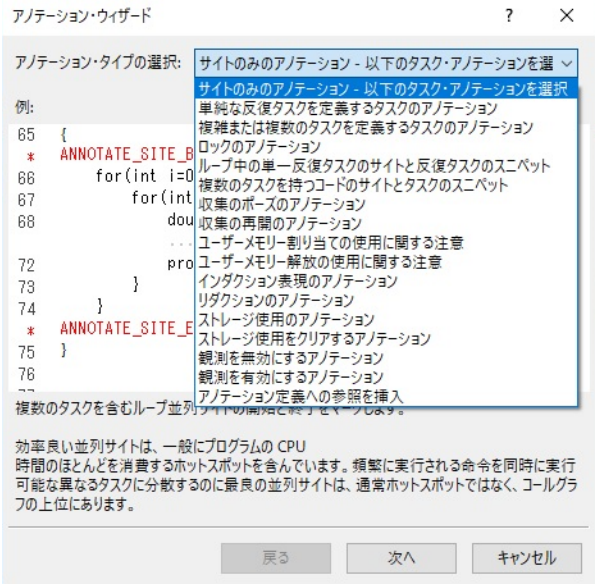
C#: Annotate.DisableCollectionPush();

収集の再開:

C/C++: ANNOTATE_DISABLE_COLLECTION_POP;

Fortran: call annotate_disable_collection_pop()

C#: Annotate.DisableCollectionPop();



© 2018 iSUS *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。

解析を始めましょう！

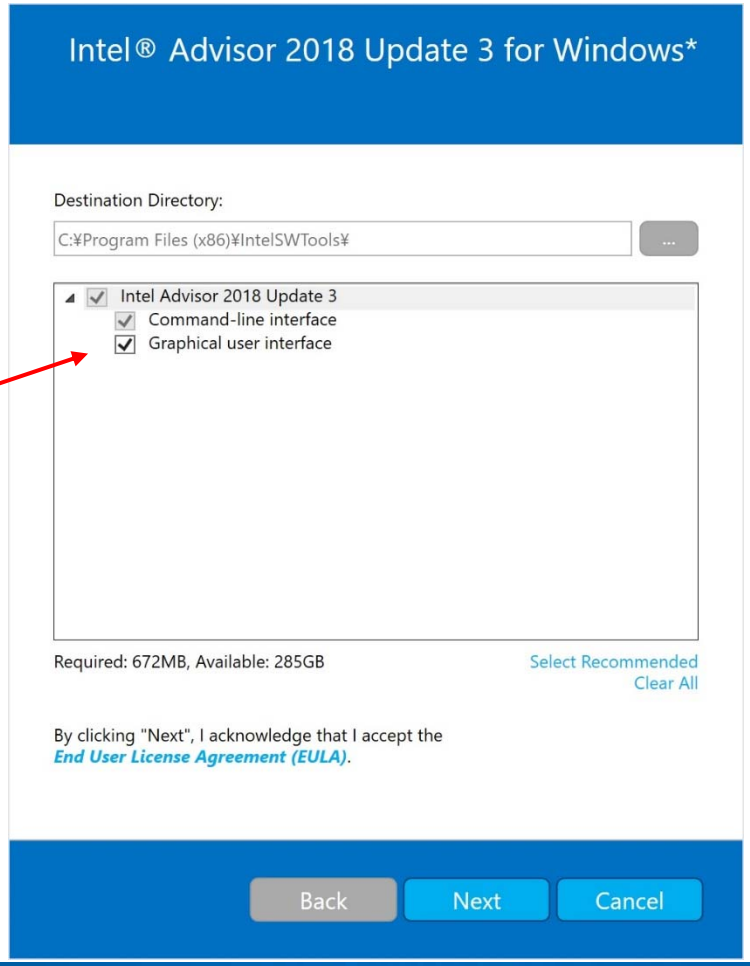
- インテル® Advisor の操作は直観的です
- アノテーション機能を除きソースを変更することなく、アプリケーションのベクトル化と並列化の候補に関連する情報を取得できます
- インテル® VTune™ Amplifier やインテル® Inspector と合わせて、アプリケーションのパフォーマンスと信頼性を向上させます

インテル® Advisor 2018 の日本語版を是非ご利用ください

© 2018 iSUS *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。

参考

コマンドライン･ツールのみをインストールする場合、Graphical user interfaceのチェックを外します



© 2018 iSUS *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。

iSUS

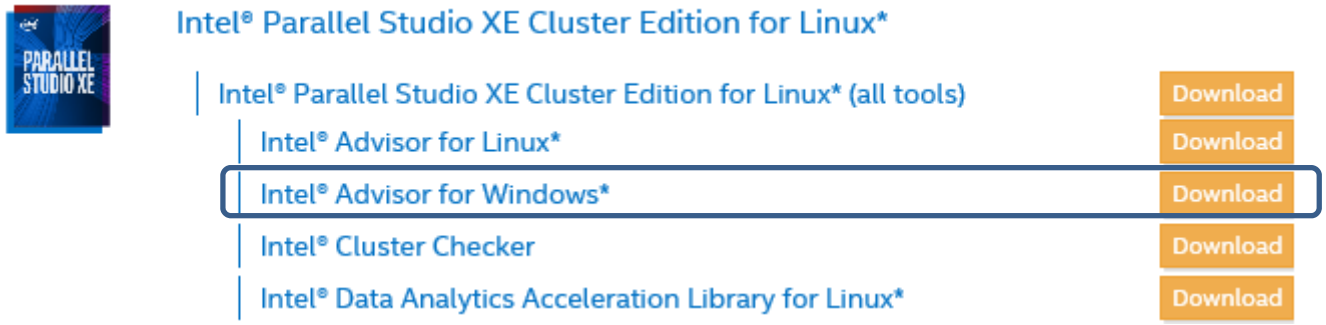
インテル® Advisor 2018 日本語版で Linux* アプリケーションを解析する手順

内容

- パッケージの入手方法
- 解析を始める前に
- Windows* バイナリーと同様の解析

© 2018 iSUS *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。

パッケージの入手方法



- インテル® Parallel Studio XE 2018 Professional/Cluster エディションの有効なライセンスを保持している場合、registrationcenter.intel.com からインテル® Advisor for Windows* をダウンロードすることができます
- パッケージをダウンロードして、インストール後、日本語パッケージを導入してください

© 2018 iSUS *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。

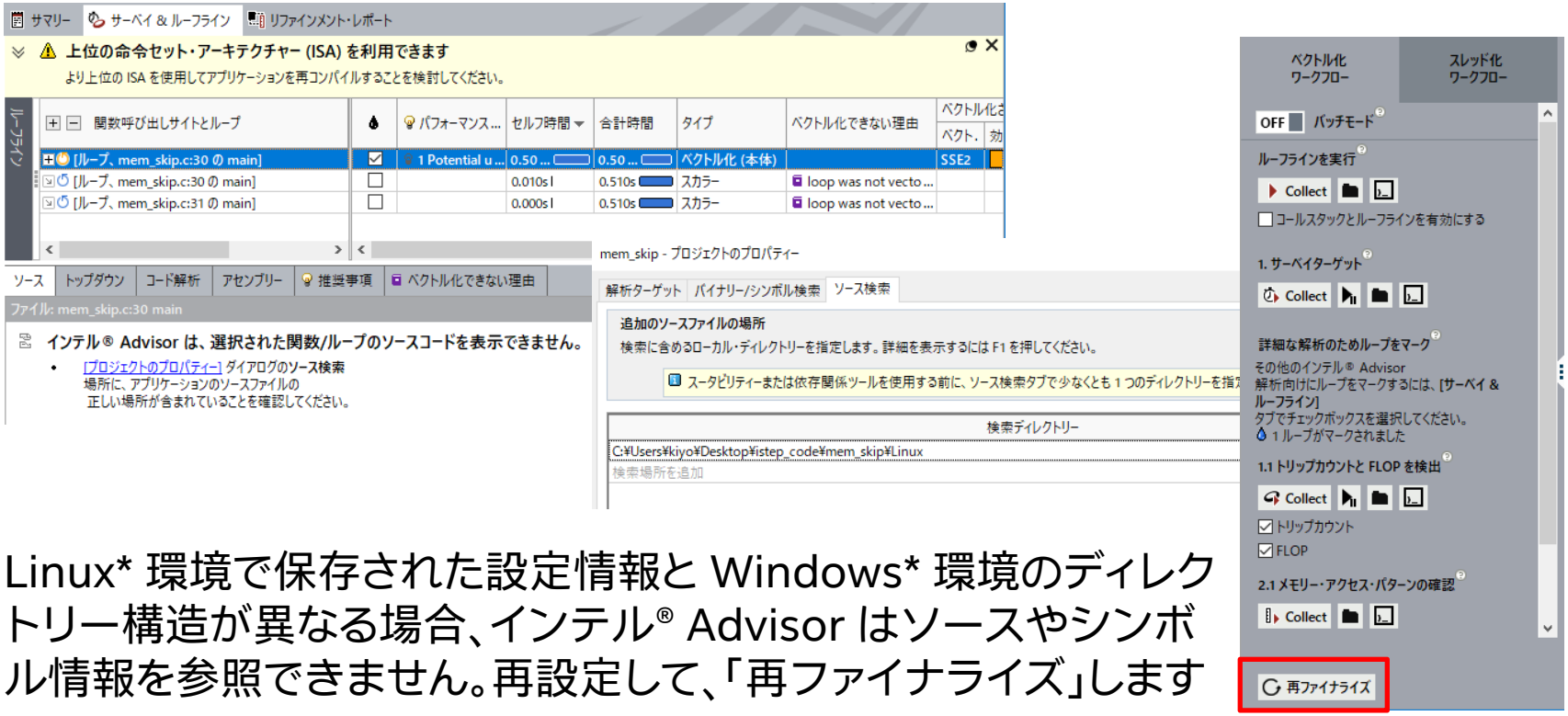
解析を始める前に:
Linux* 環境からファイルをコピー

- Linux* 環境で収集したインテル® Advisor 2018 の結果ファイルとアプリケーションのソース/バイナリーを Windows* ローカルにコピーします

- Linux* と Windows* でのインテル® Advisor 2018 の結果ファイルの構成は同一であるため、区別しやすいディレクトリー名で保存するとよいでしょう

© 2018 iSUS *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。

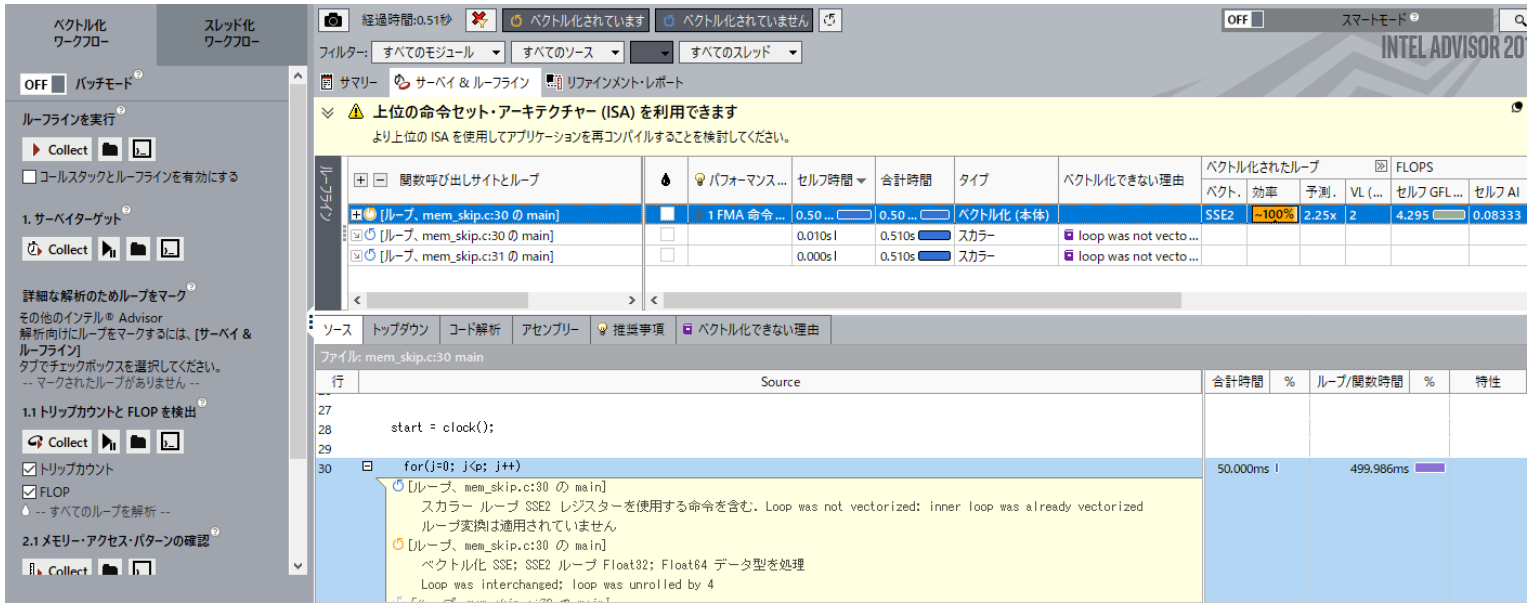
解析を始める前に: 設定の変更



Linux* 環境で保存された設定情報と Windows* 環境のディレクトリー構造が異なる場合、インテル® Advisor はソースやシンボル情報を参照できません。再設定して、「再ファイナライズ」します

© 2018 iSUS *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。

Windows* バイナリーと同様の解析



ただし、Collect ボタンを押してデータを再収集することはできません

© 2018 iSUS *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。

なぜ Linux* 環境での日本語版は提供されないのか?

このような操作が可能であるため、
早期リリースとパッケージの保守性の観点から、
インテル® Advisor 2018 は、
Windows* 版のみがリリースされています

戻る

© 2018 iSUS *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。