このセッションには音声が入っています。
スピーカーの音量に注意してください。

スライドの画面をクリックするか、
プレイヤーの再生ボタンを押すとセッションが始まります。

音量に注意してください。調整が終わったら、スタートしてください。

このスライドに音声はありません

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

インテル® Xeon Phi™ コプロセッサー Code Modernization セミナー

インテル® Xeon Phi™ コプロセッサー向けソフトウェア開発

Part 2 ソフトウェア開発ツール (約 17 分)

2015 年 10 月

内容

- インテル® Xeon Phi™コプロセッサーに適したアプリケーション領域とプログラミング環境 (Part 1)
- インテル® Xeon Phi™ コプロセッサーのソフトウェア開発をサポートする開発ツール (Part 2)
- インテル® Xeon Phi™ コプロセッサーのプログラミング (Part 3)

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

インテル® MIC 用ソフトウェア開発エコシステム[1]

	オープンソース	商用製品
コンパイラー、実行環境	gcc (カーネルビルドのみ、アプリケーション非対応)、Python	インテル® C++ コンパイラー、インテル® Fortran コンパイラー、MYO、CAPS* HMPP* コンパイラー、ScaleMP*
デバッガー	gdb	インテル® デバッガー、Rogue Wave* TotalView*、Allinea* DDT
ライブラリー	TBB[2]、MPICH2、FFTW、NetCDF	NAG*、インテル® MKL、インテル® MPI、OpenMP* (インテル® コンパイラー)、インテル® Cilk™ Plus (インテル® コンパイラー)、Co-Arrray Fortran (インテル)、Rogue Wave* IMSL、インテル® IPP
プロファイルと解析ツール		インテル® VTune™ Amplifier XE、インテル® Trace Analyzer、インテル® Trace Collector、インテル® Inspector XE
ワークロード・スケジューラー		Altair* PBS Professional、Adaptive* Computing Moab

1 – すべて発表済み。インテルはこのほかにも開発中のものがあることを言及していますが、まだ正式に発表されていません。
2 – インテル® TBB のサポートはインテルが提供。

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

インテル® ソフトウェア開発製品

インテル® Parallel Studio XE Professional Edition 高いパフォーマンス

- C++ および Fortran コンパイラー、インテル® MKL / インテル® IPP ライブラリーと解析ツール
- IA ベース・マルチコア・ノード上の Windows* および Linux* 開発者向け

インテル® Parallel Studio XE Cluster Edition 分散パフォーマンス

- MPI クラスターツールと C++ および Fortran コンパイラー、インテル® MKL / インテル® IPP ライブラリーと解析ツール
- IA ベースのクラスター上の Windows* および Linux* 開発者向け

http://www.isus.jp/article/intel-software-dev-products/intel-parallel-studio-xe/

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

インテル® ソフトウェア開発製品
(インテル Xeon® Phi™ コプロセッサー対応ツール)

Professional Edition

- インテル® Advisor XE
- ✓ インテル® C++ コンパイラー
- ✓ インテル® Fortran コンパイラー
- ✓ インテル® MKL
- インテル® IPP
- ✓ インテル® TBB
- インテル® Inspector XE
- ✓ インテル® VTune ™ Amplifier XE

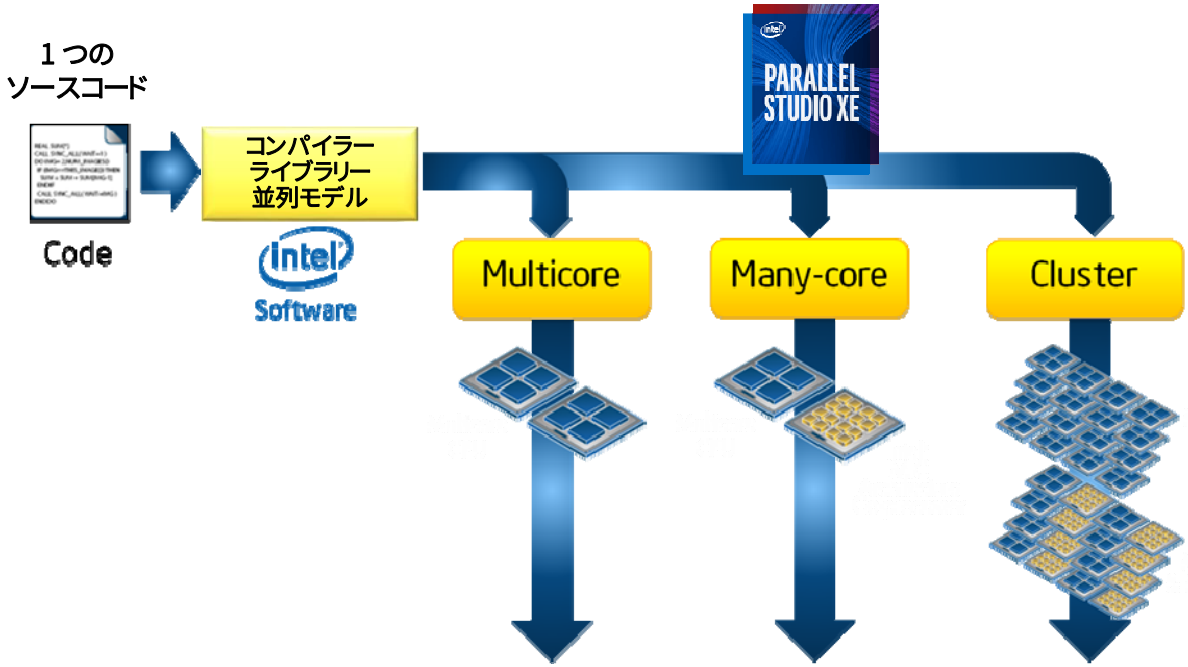
Cluster Edition

- インテル® Advisor XE
- ✓ インテル® C++ コンパイラー
- ✓ インテル® Fortran コンパイラー
- ✓ インテル® MKL
- インテル® IPP
- ✓ インテル® TBB
- インテル® Inspector XE
- ✓ インテル® VTune ™ Amplifier XE
- ✓ インテル® MPI ライブラリー
- ✓ インテル® Trace Analyzer/Collector

- 対応は Linux* 版と Windows* 版
- 詳細は各製品のリリースノートやドキュメント等をご覧ください

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

インテル® ソフトウェア開発製品の活用



共通のソースコードから複数のプラットフォームへの対応が可能

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

インテル® Parallel Studio XE Composer Edition

- ヘテロジニアス・コンパイルを提供するインテル® コンパイラー
- インテル® MIC 対応数値演算ライブラリー (インテル® MKL)
- インテル® MIC 対応インテル® デバッガー
- 並列化モデル言語拡張およびライブラリー

† 注意: インテル® IPP は現在インテル® Xeon Phi™ コプロセッサー非対応

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

インテル® Cilk™ Plus : cilkplus.org

簡単な C / C++ 構文

コンパイラーは、並列化という目的を “認識” し、支援する

インテル® TBB のように、実績ある M.I.T. の Cilk 仕様がベース

- 3 つのキーワード
- 並列タスク
- ハイパーオブジェクト
- 配列表記 (アレイ・ノーテーション)
- SIMD 対応関数
- プラグマ SIMD

http://www.isus.jp/article/intel-cilk-plus/

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

インテル® Cilk™ Plus の例

インテル® Cilk™ Plus の並列モデルは、一分で説明できます

- 3 つのキーワードによる並列処理:
  - cilk_spawn
  - cilk_sync
  - cilk_for

インテル® Cilk™ Plus: オープン仕様

- 並列プログラミングの発展のため、オープンソース化されました。成果物として、gcc などがあります

```
// C 言語での並列タスクの呼び出し
cilk_for (int i=0; i<n; ++i){
   Foo(a[i]);
}
```

```
// 再帰的な並列スポーン
// C 言語でのフィボナッチ数列の計算
int fib (int n) {
    if (n < 2) return 1;
    else {
      int x, y;
      x = cilk_spawn fib(n-1);
      y = fib(n-2);
      cilk_sync;
      return x + y;
    }
}
```

インテル® Cilk™ Plus はマルチコアとメニーコア・プログラミングに対応

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

インテル® スレッディング・ビルディング・ブロック threadingbuildingblocks.org

- 並列化向けに C++ を拡張
- 2006 年に登場
- 最も広く使用されている並列抽象化

C++ ライブラリー

並列タスク

並列アルゴリズム

コンカレント・コンテナー

同期プリミティブ

スケーラブルなメモリー割り当て機能

http://www.isus.jp/article/intel-tbb

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

インテル® スレッディング・ビルディング・ブロック (インテル® TBB)

C++ の開発者が最も多く用いる並列処理の抽象化

- C++ テンプレート・ライブラリー
- スケーラブルなメモリー・アロケーション
- スレッド・セーフなパイプライン処理
- コンカーレント・コンテナ
- 抽象化された並列アルゴリズム
- 豊富な同期プリミティブ
- ロードバランス
- ワーク・スティールのタスク・スケジュール

```
//Parallel function invocation example, in
C++, //using TBB:
```

```
parallel_for (0, n,
   [=](int i) {
   Foo(a[i]);
});
```

インテルは、インテル® TBB のサポートと支援ツールの主要提供者として、インテル® TBB オープンソース・プロジェクトに参加し貢献を継続します

インテル® TBB はメニー・コアにもマルチ・コアにも対応

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

インテル® TBB とインテル® Cilk™ Plus は優れた組み合わせ

ベクトル並列化

- インテル® Cilk™ Plus にはベクトル並列向けの 2 つの構文がある
  - 配列表記 (アレイ・ノーテーション)
  - #pragma simd
- インテル® TBB はベクトル並列をコンパイラーに依存している
  - インテル® TBB + #pragma simd は魅力的な組み合わせ

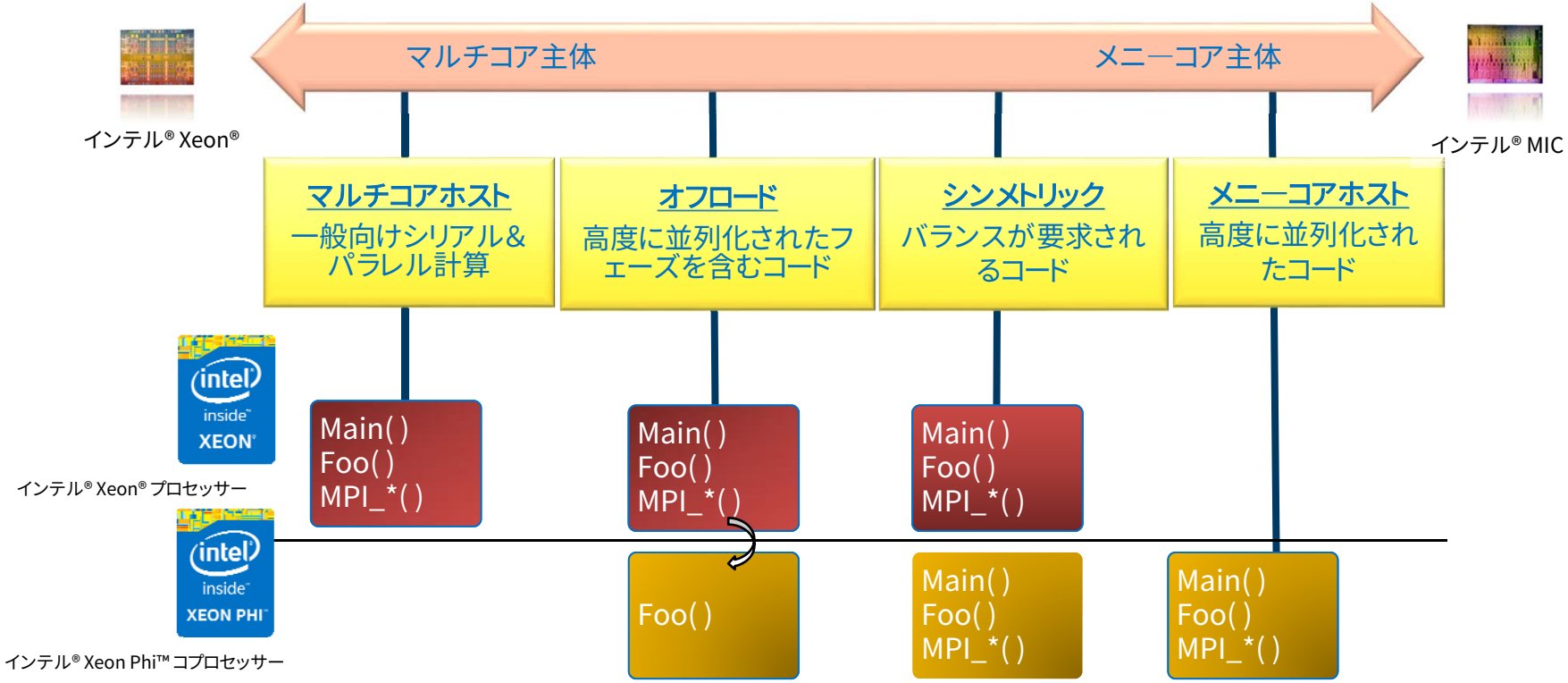
スレッド並列化

- インテル® Cilk™ Plus は厳格な fork-join 型言語
  - 制限により空間を保証
- インテル® TBB では任意のタスクグラフを使用可能
  - 柔軟性によりスレッド並列化を支援

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

インテル® MIC 対応インテル® MPI ライブラリー

◆ 柔軟なプログラミング / 実行モデルを提供



©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

インテル® MIC 対応インテル® MPI ライブラリー – 関連記事

インテル® Xeon Phi™ コプロセッサー・システムにおけるインテル® MPI ライブラリーの使用
http://www.isus.jp/article/mic-article/using-the-intel-mpi-library-on-intel-xeon-phi-coprocessor-systems/

インテル® MPI を使用した並列化
http://www.isus.jp/article/mic-article/parallelization-using-intel-mpi/

インテル® Xeon® プロセッサー・ベースのホストとインテル® Xeon Phi™ コプロセッサー間の MPI 通信の改善
http://www.isus.jp/article/mic-article/improve-mpi-com-between-xeon-host-and-xeon-phi/

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

インテル® MKL – ヘテロジニアス対応

- ヘテロジニアス対応インテル® MKL には、インテル® MIC アーキテクチャーに最適化された一般的なドメインが含まれる: BLAS、LAPACK、FFT、VML、VSL およびインテル® MKL サービス関数
  - インテル® MKL の関数名を変更する必要はない
- ホストで実行された C/C++ や Fortran コードを、インテル® MIC で実行するように指示

■ 明示的なオフロード

```
void foo(){
 float *A, *B, *C;              /* 配列 */
 #pragma offload target(mic)
  in(transa,transb,N,alpha,beta) in(A:length(matrix_elements))
  in(B:length(matrix_elements)) in(C:length(matrix_elements))
  out(C:length(matrix_elements)alloc_if(0))
 sgemm(&transa, &transb, &N,&N,&N,&alpha,A,&N,B,&N,&beta,C, &N);
}
```

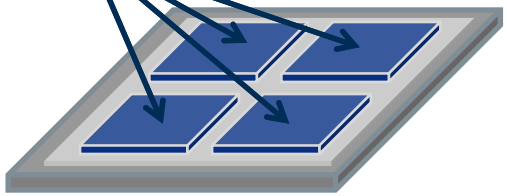
■ 自動オフロード機能 ・・・ 計算量に応じて自動的にホストとターゲットを使用する
- 環境変数 (MKL_MIC_ENABLE=1) を設定する

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

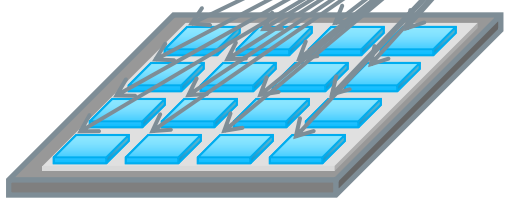
インテル® MKL による自動オフロード

```
void foo()  /* インテル® マス・カーネル・ライブラリー */
{
      float *A, *B, *C; /* Matrices */
      sgemm(&transa, &transb, &N, &N, &N, &alpha, A, &N, B, &N, &beta, C, &N);
}
```

暗黙の自動オフロード機能により、オフロード インテル® MKL ライブラリーとリンクするだけで、コードを変更する必要はありません



インテル® Xeon® プロセッサー



インテル® Xeon Phi™ コプロセッサー

インテルのハイパフォーマンスなマス・カーネル・ライブラリーは、マルチコアとメニーコア・プログラミングに対応

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

並列化手法の比較

IA の利点: 多様な開発手法から選択可能

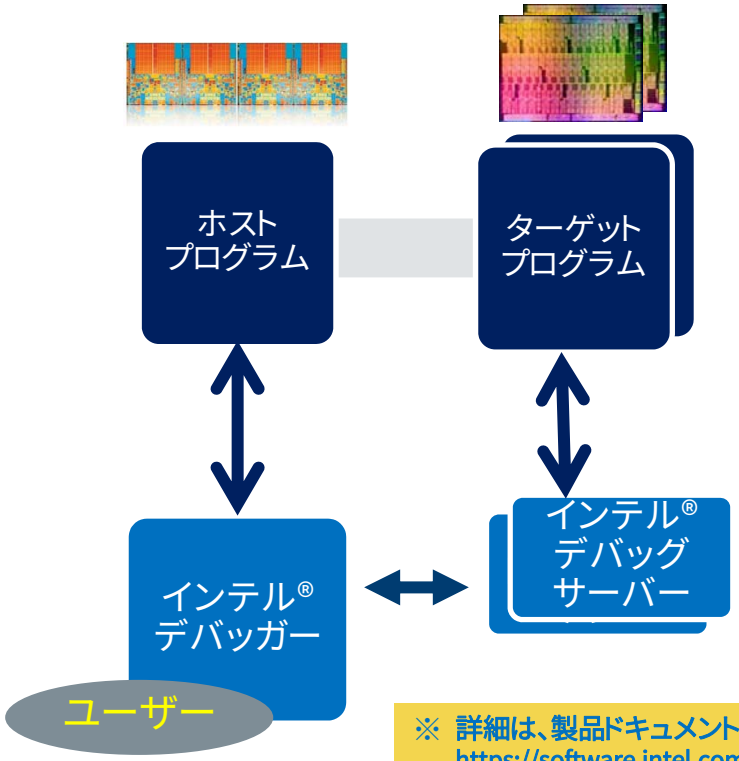
並列化手法	ベクトル化手法
インテル® MKL、 インテル® MPI ライブラリー	インテル® MKL
	自動ベクトル化
OpenMP*	半自動ベクトル化: #pragma (vector, ivdep, simd) #pragma omp simd (OpenMP*)
インテル® TBB、 インテル® Cilk™ Plus	配列表記: インテル® Cilk™ Plus
	C/C++ Vector クラス (F32vec16, F64vec8)
	OpenCL*
Pthreads* / Win32 スレッド	組み込み関数

容易性

詳細な制御

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

インテル® デバッガー (IDB)



- ホストとターゲットを同時にデバッグ
- ホストのアプリケーションのデバッグを開始すると、ターゲットのアプリケーションは自動的にデバッグモードに入る
- ホストとターゲットのデバッグには、ホスト側のデバッガーを利用する
- デバッガーは、ホストとターゲットのプログラムを同期して、停止/再開できる
- 双方で C++ と FORTRAN を完全にサポート
- 将来のバージョンでは、1 つの GUI の中に 1 つの仮想アプリケーションのみを表示できるようにする
- 1 つ以上のオフロードカードをカバーするように拡張可能

※ ターゲットとはインテル® Xeon Phi™ コプロセッサーを指す

※ 詳細は、製品ドキュメントまたは以下のサイトをご参照ください。
https://software.intel.com/en-us/articles/debugging-intel-xeon-phi-applications-on-linux-host

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

インテル® デバッガー (IDB) - 関連記事

Windows* ホスト上でインテル® Xeon Phi™ コプロセッサー向けアプリケーションをデバッグする
http://www.isus.jp/article/mic-article/debugging-intel-xeon-phi-applications-on-windows-host/

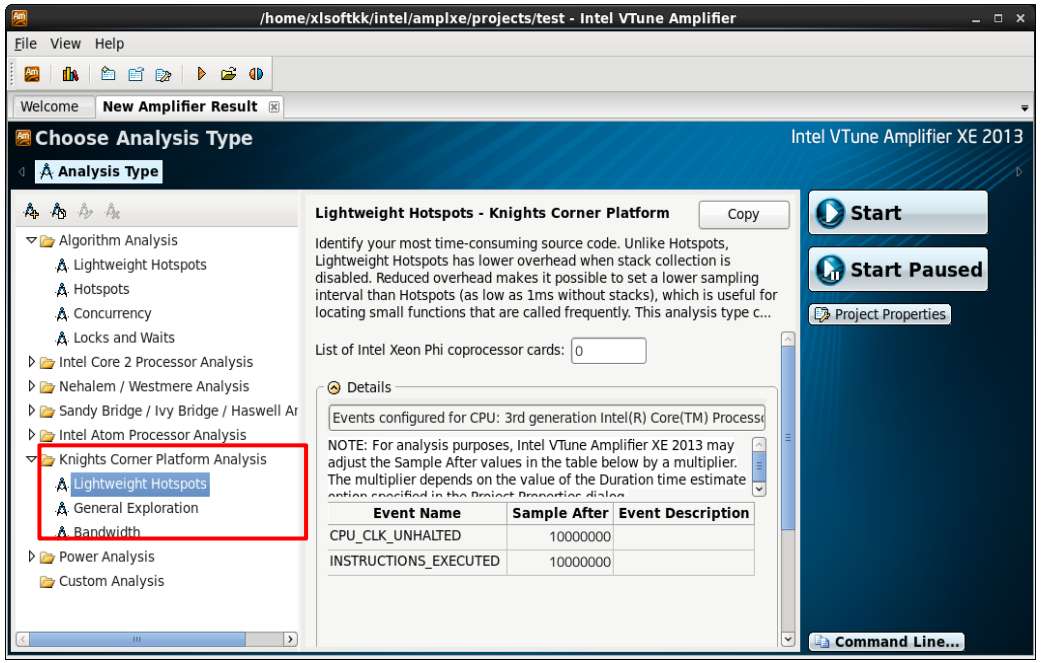
Linux* ホスト上でインテル® Xeon Phi™ コプロセッサー向けアプリケーションをデバッグ
http://www.isus.jp/article/mic-article/debugging-intel-xeon-phi-applications-on-linux-host/

インテル® Xeon Phi™ コプロセッサーでのデバッグ: 使用例
http://www.isus.jp/article/mic-article/debugging-on-intel-xeon-phi-coprocessor-use-case-overview/

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

インテル® VTune™ Amplifier XE

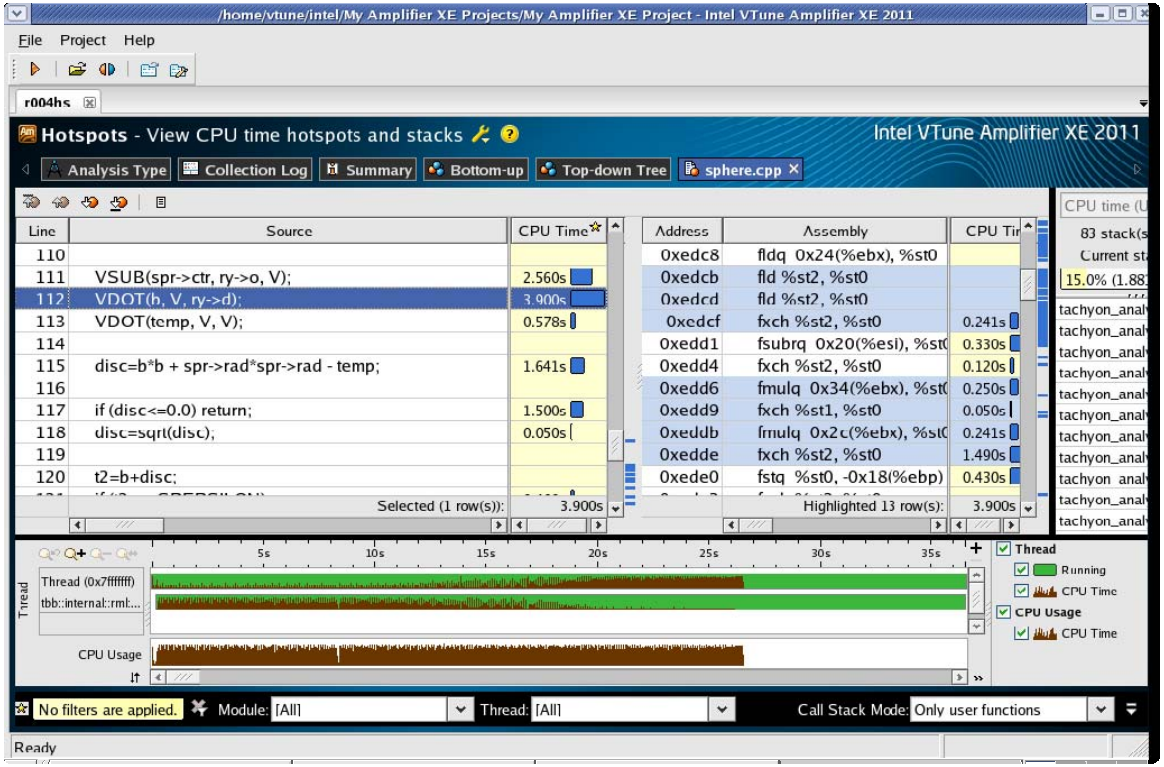
インテル® VTune™ Amplifier XE パフォーマンス・プロファイラーは、
オフロード実行およびネイティブ実行モデルにおけるインテル® MIC のパフォーマンスを解析



http://www.isus.jp/article/intel-software-dev-products/intel-vtune-amplifier-xe/

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

インテル® VTune™ Amplifier XE
ソースコードの該当箇所を表示



©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

インテル® VTune™ Amplifier XE – 関連記事

クライアント・マシンからサーバー上のインテル® Xeon Phi™ コプロセッサーで動作するアプリケーションのプロファイルを行う
http://www.isus.jp/article/mic-article/practice-an-example-of-profiling-app-on-xeon-phi/

インテル® VTume™ Amplifier XE 2015 におけるインテル® Xeon Phi™ コプロセッサー向けコールスタック・イベントベース・サンプリング
http://www.isus.jp/article/mic-article/event-based-sampling-with-call-stack-for-intel-xeon-phi-is-coming-in-vtune/

インテル® Xeon Phi™ コプロセッサー向けの最適化とパフォーマンス・チューニング – パート 1: 最適化の基本
http://www.isus.jp/article/mic-article/optimization-and-performance-tuning-for-intel-xeon-phi-coprocessors-part-1-optimization/

インテル® Xeon Phi™ コプロセッサー向けの最適化とパフォーマンス・チューニング – パート 2: ハードウェア・イベントの理解と使用
http://www.isus.jp/article/mic-article/optimization-and-performance-tuning-for-intel-xeon-phi-coprocessors-part-2-understanding/

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

まとめ

インテルはプロセッサーとコプロセッサー向けに標準化された共通の方法で並列化等の最適化を行うツールを提供しており、将来にわたって、ソフトウェア資産の継続的な利用を目指す

- 既存の開発環境と共通のツール
- 既存のプログラミング・モデル
- 単一のソースコードから 2 つのアーキテクチャーに対応
- スケーラブル

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

日本語情報サイト

http://www.isus.jp/article/idz/mic-developer/

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

関連書籍

Structured Parallel Programming: Patterns for Efficient Computation

著者 Michael McCool, James Reinders ,Arch Robison 出版日: 2012 年 7 月 9 日 | ISBN: 978-0-124159938

『構造化並列プログラミング: 効率良い計算を行うためのパターン』

著者　マイケル・マックール／アーク・D・ロビソン／ジェームス・レインダース（共著）

訳者　菅原 清文／エクセルソフト株式会社（共訳）| ISBN　978-4-87783-305-3

Intel® Xeon Phi™ Coprocessor High Performance Programming

著者 Jim Jeffers , James Reinders 出版日: 2013 年 3 月 | ISBN: 978-0-124104143

『インテル® Xeon Phi™ コプロセッサー ハイパフォーマンス・プログラミング』

著者　ジェームス・レインダース/アーク・D・ロビソン（共著）

訳者　菅原 清文／エクセルソフト株式会社（共訳）| ISBN　978-4-87783-332-9

High Performance Parallelism Pearls

著者 Jim Jeffers , James Reinders　出版日: 2014 年 11 月

簡単にインテル® Xeon Phi™ コプロセッサー・ファミリーの優れた並列性を利用してコードを実行できるため、最適化に集中し、ハイパフォーマンスを実現することが可能です。並列処理を細かくチューニングすることで、正しいアプリケーションを正しく効率良いアプリケーションにすることができます。インテル コーポレーションの並列プログラミング・エバンジェリストである James Reinders とインテル コーポレーションのエンジニアである Jim Jeffers により執筆された最新の書籍は、69 人の専門家の実際の経験を基に、インテルのマルチコアおよびメニーコア・プロセッサーを最大限に利用するための創意工夫を紹介しています。

©2015 Intel Corporation. 無断での引用、転載を禁じます。* 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。

intel®

法務上の注意書きと最適化に関する注意事項

本資料の情報は、現状のまま提供され、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。製品に付属の売買契約書『Intel's Terms and Conditions of Sale』に規定されている場合を除き、インテルはいかなる責任を負うものではなく、またインテル製品の販売や使用に関する明示または黙示の保証 (特定目的への適合性、商品性に関する保証、第三者の特許権、著作権、その他、知的財産権の侵害への保証を含む) をするものではありません。
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。

© 2015 Intel Corporation. 無断での引用、転載を禁じます。Intel、インテル、Intel ロゴ、Cilk、Intel Xeon Phi、VTune、Xeon 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。

最適化に関する注意事項

インテル® コンパイラーは、互換マイクロプロセッサー向けには、インテル製マイクロプロセッサー向けと同等レベルの最適化が行われない可能性があります。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® SSE2)、インテル® ストリーミング SIMD 拡張命令 3 (インテル® SSE3)、ストリーミング SIMD 拡張命令 3 補足命令 (SSSE3) 命令セットに関連する最適化およびその他の最適化が含まれます。インテルでは、インテル製ではないマイクロプロセッサーに対して、最適化の提供、機能、効果を保証していません。本製品のマイクロプロセッサー固有の最適化は、インテル製マイクロプロセッサーでの使用を目的としています。インテル® マイクロアーキテクチャーに非固有の特定の最適化は、インテル製マイクロプロセッサー向けに予約されています。この注意事項の適用対象である特定の命令セットの詳細は、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。

改訂 #20110804