

高速なコードを 素早く開発

インテル® Parallel Studio XE 2017

高速なコードを素早く開発

- インテル® Parallel Studio XE
 - 設計、ビルド、検証、チューニング
 - C++, C, Fortran, Python*, Java*
 - 標準規格に基づく並列モデル: OpenMP*, MPI、インテル® TBB
- バージョン 2017 の主な機能
 - インテル® Distribution for Python* とインテル® VTune™ Amplifier XE により、Python* アプリケーションのパフォーマンスを向上
 - インテル® MKL とインテル® DAAL により、インテル® アーキテクチャー上でディープラーニングを高速化
 - インテル® VTune™ Amplifier XE とインテル® Trace Analyzer & Collector のスナップショット機能により、アプリケーションのパフォーマンスを迅速に診断
 - 次世代のプラットフォームでスケーリング (最新のインテル® Xeon Phi™ プロセッサを含む) インテル® AVX-512、高帯域メモリ、コンパイラおよび解析ツールの明示的なベクトル化を最適化

AVX-512
PERFORMANCE
MPI PYTHON
DATA ANALYTICS XEON
XEON PHI
MACHINE LEARNING
VECTORIZATION
THREADING



<http://intel.ly/perf-tools> (英語)

最適化に関する注意事項

© 2016 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

プロファイル、解析、
アーキテクチャ

インテル® Inspector
メモリー/スレッドのチェック

インテル® VTune™ Amplifier XE
パフォーマンス・プロファイラー

パフォーマンス・
ライブラリー

インテル® DAAL
データ解析/マシンラーニング向けに最適化済み

インテル® MKL
工学、科学、金融系アプリケーション向けに最適化されたルーチン

インテル® Advisor

ベクトル化の最適化とスレッドのプロトタイプ生成

インテル® Cluster Checker

クラスター診断エキスパート・システム

インテル® Trace Analyzer & Collector
MPI プロファイラー

クラスターツール

インテル® MPI ライブラリー

インテル® IPP
画像、信号、圧縮ルーチン

インテル® TBB
タスクベースの並列 C++ テンプレート・ライブラリー

インテル® C/C++ および Fortran コンパイラー

インテル® Distribution for Python*
パフォーマンスを引き出すスクリプト

最適化に関する注意事項

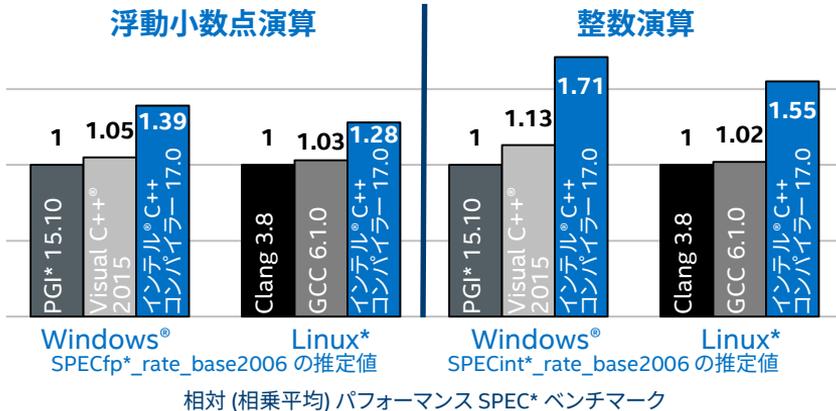
© 2016 Intel Corporation. 無断での引用、転載を禁じます。

* 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。

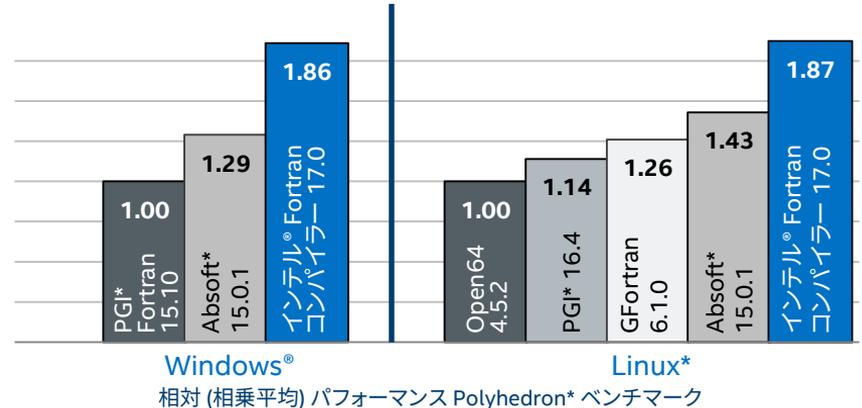


Windows®/Linux* 上でアプリケーション・パフォーマンスを向上 インテル® C++ および Fortran コンパイラー

インテル® C++ コンパイラーによる優れた C++
アプリケーション・パフォーマンス — Windows®/Linux*
(数値が大きいほど高性能)



インテル® Fortran コンパイラーによる優れた Fortran
アプリケーション・パフォーマンス — Windows®/Linux*
(数値が大きいほど高性能)



システム構成: ハードウェア: インテル® Xeon® プロセッサ E3-1245 v5 @ 3.50GHz、ハイバースレディング有効、ターボブースト有効、32GB RAM、ソフトウェア: インテル® Xeon® プロセッサ E5-2680 v5 @ 2.50GHz、256GB RAM、ハイバースレディング有効、ソフトウェア: インテル® C++ コンパイラー 17.0、Microsoft® C/C++ 最適化コンパイラー 19.00.23918 (x86/x64)、GCC 6.1.0、PGI* 15.10、Clang/LVM 3.8、Linux OS: Red Hat® Enterprise Linux® Server 7.1 (Maipo)、カーネル 3.10.0-229.el7.x86_64、Windows OS: Windows® 10 Pro (10.0.10240 N/A Build 10240)、SPEC* ベンチマーク (www.spec.org)、SPECint* ベンチマークでは、Visual C++® コンパイラーとインテル® コンパイラーで SmartHeap 11.3 を使用。

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・システム、コンピュータ、ソフトウェア、操作、機能に基づいて行われたものです。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。

* 其他の社名、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。ベンチマークの出典: インテル コーポレーション

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用に同等の最適化を行えないことがあります。これは、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依存の最適化は、インテル® マイクロプロセッサ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。注意事項の改訂 #20110804

システム構成: ハードウェア: インテル® Xeon® プロセッサ E3-1245 v5 @ 3.50GHz、ハイバースレディング有効、ターボブースト有効、32GB RAM、ソフトウェア: インテル® Fortran コンパイラー 17.0、Absort* 15.0.1、PGI* Fortran 15.10、Windows® OS: Windows® 10 Pro (10.0.10240 N/A Build 10240)、Polyhedron* Fortran ベンチマーク (www.fortran.uk)、Windows® コンパイラー・オプション: Absort* -m64 -O5 -speed_math=10 -fast_math -march=core -xINTEGER -stack=0x8000000、インテル® Fortran コンパイラー: /fast /Qparallel /QxCORE-AVX2 /nostandard-realloc-ifs /link /stack=6400000、PGI* Fortran -fastsse -Mnroll=m4 -Mipa-fast:inline -Mconcur=numa、Linux* コンパイラー・オプション: Absort* -m64 -mavx -O5 -speed_math=10 -march=core -xINTEGER、GFortran: -Ofast -mpmath=sse -fipo -march=native -funroll-loops -free-parallelize-loops=4、インテル® Fortran コンパイラー: -fast -parallel -xCORE-AVX2 -nostandard-realloc-ifs、PGI* Fortran: -fast -Mipa-fast:inline -Msmartalloc -Mpreaved -Mstack_arrays -Mconcur=bind-Open64 -march=auto -Ofast -mso -ipo、

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・システム、コンピュータ、ソフトウェア、操作、機能に基づいて行われたものです。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。

* 其他の社名、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。ベンチマークの出典: インテル コーポレーション

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用に同等の最適化を行えないことがあります。これは、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依存の最適化は、インテル® マイクロプロセッサ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。注意事項の改訂 #20110804

最適化に関する注意事項

© 2016 Intel Corporation. 無断での引用、転載を禁じます。

* 其他の社名、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。



各エディションの概要

インテル® Parallel Studio XE 2017



		Composer Edition	Professional Edition	Cluster Edition
ビルド	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® Distribution for Python*	✓	✓	✓
	インテル® MKL – 高速な数学ライブラリー	✓	✓	✓
	インテル® IPP – 画像、信号、およびデータ処理	✓	✓	✓
	インテル® TBB – スレッド・ライブラリー	✓	✓	✓
	インテル® DAAL – マシンラーニングとデータ解析	✓	✓	✓
解析	インテル® VTune™ Amplifier XE – パフォーマンス・プロファイラー		✓	✓
	インテル® Advisor – ベクトル化の最適化とスレッドのプロトタイプ生成		✓	✓
	インテル® Inspector – メモリー/スレッドのデバッガー		✓	✓
スケール	インテル® MPI ライブラリー – メッセージ・パッシング・インターフェイス・ライブラリー			✓
	インテル® Trace Analyzer & Collector – MPI チューニングと解析			✓
	インテル® Cluster Checker – クラスター診断エキスパート・システム			✓
	ログウェーブ IMSL* ライブラリー – Fortran 数値解析	バンドルまたは アドオン	アドオン	アドオン

フローティング・ライセンスおよびアカデミック・ライセンスを含むその他の構成については、<http://intel.ly/perf-tools> (英語) を参照してください。

最適化に関する注意事項

© 2016 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。



最新の標準規格、オペレーティング・システム、 プロセッサのサポート

C11 および C++14 言語標準のサポートを拡張

- メモリー解放時のサイズ指定
- constexpr 制限の緩和
- 可変テンプレート
- 数値区切りとしての一重引用符

オペレーティング・システム

- Windows® 7 - 10、Windows Server® 2008 - 2012
- Debian* 7.0/8.0、Fedora* 23/24、Red Hat* Enterprise Linux* 6/7、SuSE* LINUX Enterprise Server 11/12、Ubuntu* 14.04 LTS/16.04 LTS/16.04
- macOS* 10.11

Fortran 2008 および Fortran 2015 暫定版 言語標準のサポートを拡張

- 暗黙形状 PARAMETER 配列
- Fortran 2008 BIND(C) 内部プロシージャ
- 名前付きブロックにおける EXIT の拡張
- ポインター初期化

最新のプロセッサ

- 最新のインテル® Xeon Phi™ プロセッサ (開発コード名: Knights Landing) とインテル® AVX-512 向けのチューニングとサポート

インテル® コンパイラー

インテル® Parallel Studio XE 2017 に含まれるインテル® コンパイラー インテル® C++ コンパイラー 17.0 とインテル® Fortran コンパイラー 17.0

共通の変更点

- 最新のインテル® プロセッサー (インテル® Xeon Phi™ プロセッサーを含む) のインテル® AVX2 およびインテル® AVX-512 命令セットをサポート
- コードの現代化に不可欠な最適化/ベクトル化レポートを拡張
- ベクトル化の制御を向上し、新しい SIMD 命令を提供する、OpenMP* 4.5 をサポート

インテル® C++ コンパイラー

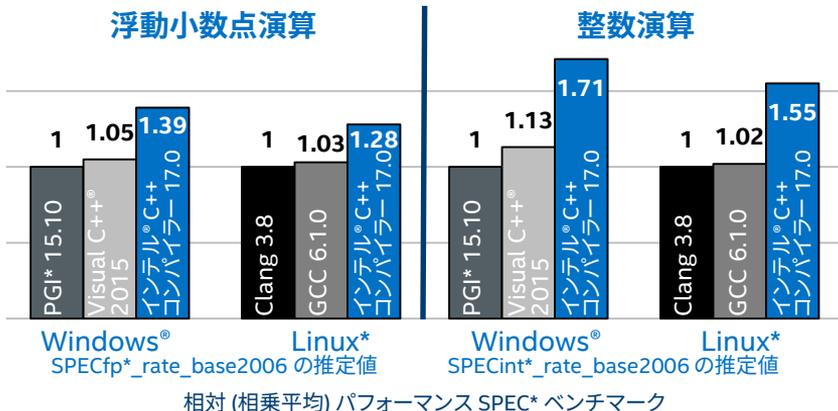
- C++ コードのベクトル化を向上する SIMD Data Layout Template (SDLT)
- 仮想関数のベクトル化
- 最新の C11、C++14 標準規格をフルサポート、C++17 の初期サポート

インテル® Fortran コンパイラー

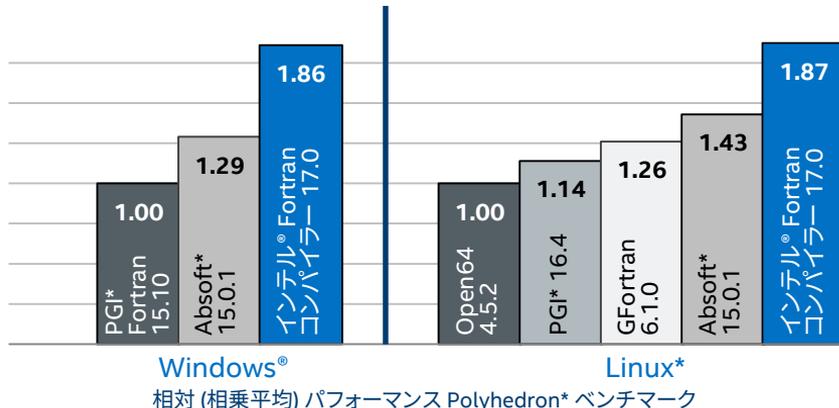
- Co-Array のパフォーマンスが大幅に向上 – Co-Array Fortran プログラムで以前のバージョンよりも最大 **2 倍スピードアップ**
- Fortran 2008 をほぼ完全にサポート
- C との互換性が向上 (Fortran 2015 暫定版の機能)

Windows®/Linux* 上でアプリケーション・パフォーマンスを向上 インテル® C++ および Fortran コンパイラー

インテル® C++ コンパイラーによる優れた C++
アプリケーション・パフォーマンス — Windows®/Linux*
(数値が大きいほど高性能)



インテル® Fortran コンパイラーによる優れた Fortran
アプリケーション・パフォーマンス — Windows®/Linux*
(数値が大きいほど高性能)



システム構成: Windows® ハードウェア: インテル® Xeon® プロセッサ E3-1245 v5 @ 3.50GHz、ハイバースレディング有効、ターボブースト有効、32GB RAM、ソフトウェア: インテル® Xeon® プロセッサ E5-2680 v5 @ 2.50GHz、256GB RAM、ハイバースレディング有効、ソフトウェア: インテル® C++ コンパイラー 17.0、Microsoft® C/C++ 最適化コンパイラー 19.00.23918 (x86/x64)、GCC 6.1.0、PGI® 15.10、Clang/LVM 3.8、Linux®: Red Hat® Enterprise Linux® Server 7.1 (Majipo)、カーネル: 3.10.0-229.el7.x86_64、Windows® OS: Windows® 10 Pro (10.0.10240 N/A Build 10240)、SPEC ベンチマーク (www.spec.org)、SPECint® ベンチマークでは、Visual C++® コンパイラーとインテル® コンパイラーで SmartHeap 11.3 を使用。

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システム、ソフトウェア、操作、構成に基づいて行われたものです。結果はこれらの要因によって異なります。製品の購入を検討する場合は、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。
*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。ベンチマークの出典: インテル® コーポレーション

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 補完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依存の最適化は、インテル® マイクロプロセッサ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。注意事項の改訂 #20110804

システム構成: ハードウェア: インテル® Xeon® プロセッサ E3-1245 v5 @ 3.50GHz、ハイバースレディング有効、ターボブースト有効、32GB RAM、ソフトウェア: インテル® Fortran コンパイラー 17.0、Abssoft® 15.0.1、PGI® Fortran 15.10、Windows® OS: Windows® 10 Pro (10.0.10240 N/A Build 10240)、Polyhedron® Fortran ベンチマーク (www.fortran.uk)、Windows® コンパイラーオプション: Abssoft: -m64 -O5 -speed_math=10 -fast_math -march=core -xINTEGR -stack=0x80000000、インテル® Fortran コンパイラー: /fast /Qparallel /QxCORE-AVX2 /nostandard-realloc-lhs /link /stack=6400000、PGI® Fortran: -fastsse -Mnroll=r4 -Mipa-fast-inline -Mconcur=numa、Linux® コンパイラーオプション: Abssoft: -m64 -mavx -O5 -speed_math=10 -march=core -xINTEGR、GFortran: -Ofast -fmpmath-sse -fIto -march=native -funroll-loops -fthree-parallelize-loops=4、インテル® Fortran コンパイラー: -fast -parallel -xCORE-AVX2 -nostandard-realloc-lhs、PGI® Fortran: -fast -Mipa-fast-inline -Msmartalloc -Mpretailed -Mstack_arrays -Mconcur=bind、Open64: -march=auto -Ofast -mso -app。

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システム、コンピュータ、ソフトウェア、操作、構成に基づいて行われたものです。結果はこれらの要因によって異なります。製品の購入を検討する場合は、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。
*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。ベンチマークの出典: インテル® コーポレーション

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 補完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依存の最適化は、インテル® マイクロプロセッサ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。注意事項の改訂 #20110804

最適化に関する注意事項

© 2016 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。

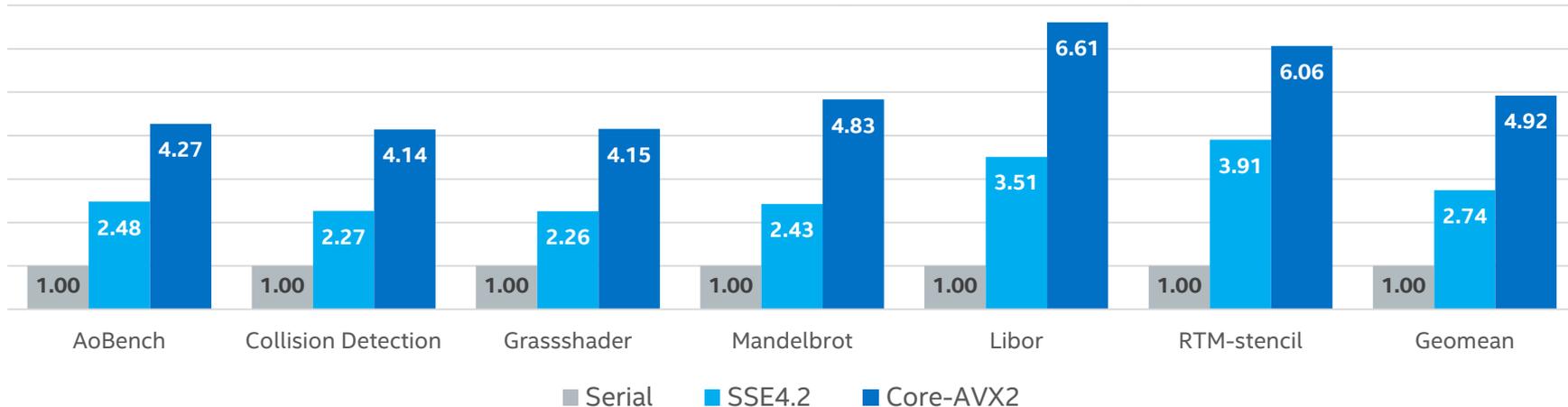


パフォーマンスを大幅に向上

OpenMP* の SIMD を使用した Intel® C++ コンパイラーの明示的なベクトル化

Intel® Xeon® プロセッサ上での SIMD のスピードアップ

正規化されたパフォーマンス・データ (数値が大きいほど高性能)



システム構成: Intel® Xeon® プロセッサ E3-1270 @ 3.50GHz Haswell† システム (4 コア、ハイパースレッディング有効)、32GB RAM、L1 キャッシュ 256KB、L2 キャッシュ 1MB、L3 キャッシュ 8MB、Windows Server® 2012 R2 Datacenter (64 ビット版)
コンパイラー・オプション: -O3 -Qopenmp -simd -QxSSE4.2 (Intel® SSE4.2 の場合) または -O3 -Qopenmp -simd -QxCORE-AVX2 (Intel® AVX2 の場合)。詳細については、<http://www.intel.com/performance> (英語) 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 Intel® マイクロプロセッサ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、機能に基づいて行われたものです。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。* 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。ベンチマークの出典: Intel コーポレーション

最適化に関する注意事項: Intel® コンパイラーでは、Intel® マイクロプロセッサ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用に同等の最適化を行えないことがあります。これには、Intel® ストリーミング SIMD 拡張命令 2、Intel® ストリーミング SIMD 拡張命令 3、Intel® 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。Intel は、他社製マイクロプロセッサに関して、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依存の最適化は、Intel® マイクロプロセッサでの使用を前提としています。Intel® 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、Intel® マイクロプロセッサ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。注意事項の改訂 #20110804

† 開発コード名

最適化に関する注意事項

© 2016 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® DISTRIBUTION FOR PYTHON*

主な機能: インテル® Distribution for Python* 2017

Python* のパフォーマンスをネイティブの速度に近づける

ハイパフォーマンスな
Python* を簡単に利用可能

- 数値/科学計算、データ解析、HPC 向けに事前にビルドされ、高速化されたディストリビューション。インテル® アーキテクチャー向けに最適化済み。
- 既存の Python* から簡単に移行可能。コード変更は不要。

複数の最適化手法により
パフォーマンスを向上

- インテル® MKL により NumPy*/SciPy*/scikit-learn のパフォーマンスを向上。
- pyDAAL によるデータ解析、インテル® TBB によるスレッド・スケジュールの強化、Jupyter* Notebook インターフェイス、Numba*、Cython。
- 最適化された mpi4py と Jupyter* Notebook により簡単にスケーリング。

インテル® アーキテクチャー
向けの最新の最適化を
素早く利用

- conda および Anaconda Cloud からディストリビューションと個別の最適化されたパッケージを利用可能。
- 最適化はメインの Python* トランクに反映される。

最適化に関する注意事項

© 2016 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。

Python* パフォーマンスを高速化するための 2 段階のアプローチ

高速な Python* ディストリビューション + パフォーマンス・プロファイル

ステップ 1: インテル® Distribution for Python* を使用

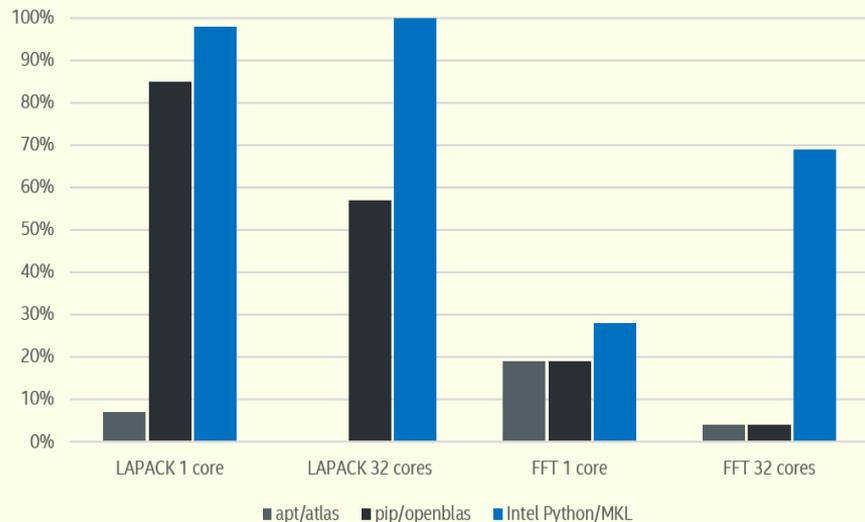
- パフォーマンスが最適化されたネイティブ・ライブラリーを利用
- 現在使用中の Python* から簡単に移行可能
- インテル® プロセッサおよびライブラリー向けの最新の最適化

ステップ 2: インテル® VTune™ Amplifier XE でプロファイル

- アプリケーション全体の実行プロファイルの詳細なサマリーを取得
- Python*/C/C++ 混在コードと拡張を自動検出し、プロファイル
- hotspot を正確に検出、行レベルの解析により迅速に賢く最適化
- インテル® Parallel Studio XE 2017 スイートのコンポーネント

IA 上でネイティブに近いパフォーマンスを達成

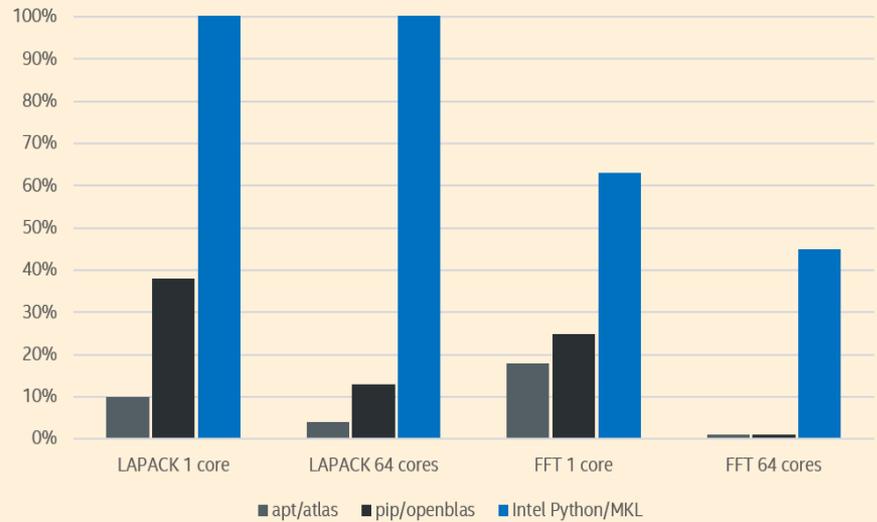
Python* Performance as a Percentage of C/Intel® MKL for Intel® Xeon® Processors (Higher is Better)



インテル® Xeon® プロセッサ

システム構成: APT/ATLAS: apt-get でインストール、Ubuntu* 16.10、Python* 3.5.2、NumPy* 1.11.0、SciPy* 0.17.0、pip/OpenBLAS: pip でインストール、Ubuntu* 16.10、Python* 3.5.2、NumPy 1.11.1、SciPy* 0.18.0。インテルの Python*: インテル® Distribution for Python* 2017。ハードウェア: インテル® Xeon プロセッサベースのシステム: インテル® Xeon® プロセッサ E5-2698 v3 @ 2.30GHz (2 ソケット、1 ソケットあたり 16 コア、HT 無効)、64GB RAM、8 DIMMS (8GB @ 2133MHz)。インテル® Xeon Phi™ プロセッサベースのシステム: インテル® Xeon Phi™ プロセッサ 7210 1.30GHz、96GB RAM、6 DIMMS (16GB @ 1200MHz)。

Python* Performance as a Percentage of C/Intel® MKL for Intel® Xeon Phi™ Product Family (Higher is Better)



インテル® Xeon Phi™ 製品ファミリー

[システム構成の詳細はこちらを参照](#)

最適化に関する注意事項

© 2016 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。

ライブラリー

インテル® MKL

インテル® DAAL

インテル® IPP

インテル® TBB

インテル® マス・カーネル・ライブラリー (インテル® MKL)

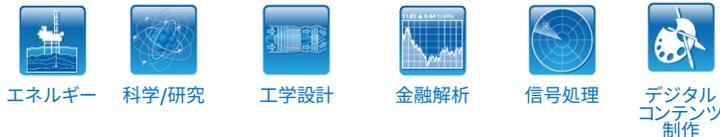
最適化に関する注意事項

© 2016 Intel Corporation. 無断での引用、転載を禁じます。

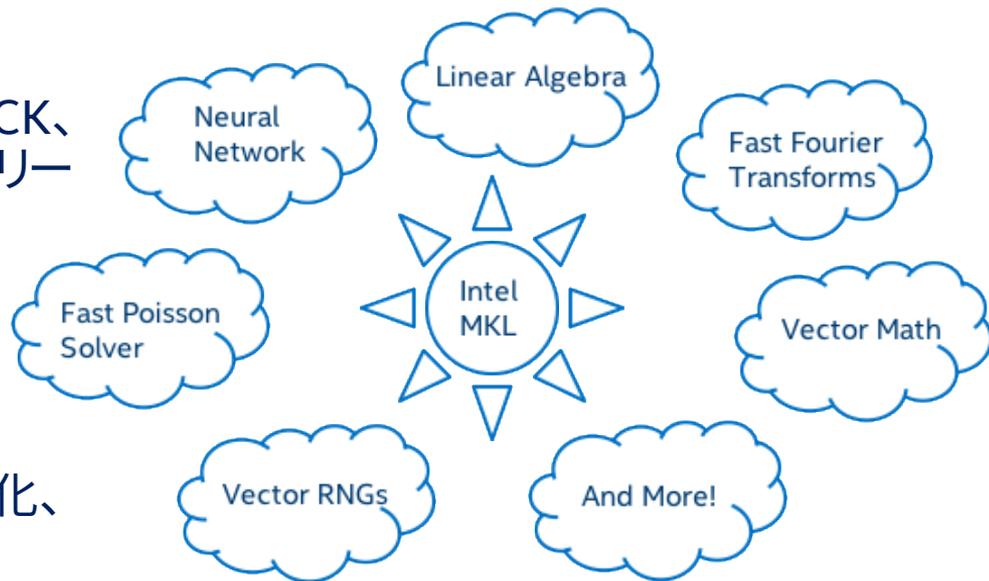
* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。



インテル® MKL



- マシンラーニング、科学、工学、金融、デザイン系アプリケーションにおける数学処理を高速化
- 密/スパース線形代数 (BLAS、LAPACK、PARDISO)、FFT、ベクトル演算、サマリー統計などの関数を含む
- ほかの数学ライブラリーから簡単に切り替えられる業界標準の API
- プロセッサのパフォーマンスを最大限に引き出すように高度に最適化、スレッド化、およびベクトル化済み



最適化に関する注意事項

© 2016 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 2017 のコンポーネント



線形代数	高速フーリエ変換 (FFT)	ベクトル演算	サマリー統計	その他	ディープ・ニューラル・ネットワーク
<ul style="list-style-type: none">• BLAS• LAPACK• ScaLAPACK• スパース BLAS• スパースソルバー• 反復法• PARDISO• クラスタースパース・ソルバー	<ul style="list-style-type: none">• 多次元• FFTW インターフェイス• クラスタ FFT	<ul style="list-style-type: none">• 三角関数• 双曲線• 指数• 対数• べき乗• 平方根• ベクトル RNG	<ul style="list-style-type: none">• 尖度• 変化係数• 順序統計量• 最小/最大• 分散/共分散	<ul style="list-style-type: none">• スプライン• 補間• 信頼区間• 高速ポアソンソルバー	<ul style="list-style-type: none">• 畳み込み• プーリング• 正規化• ReLU• ソフトマックス

最適化に関する注意事項

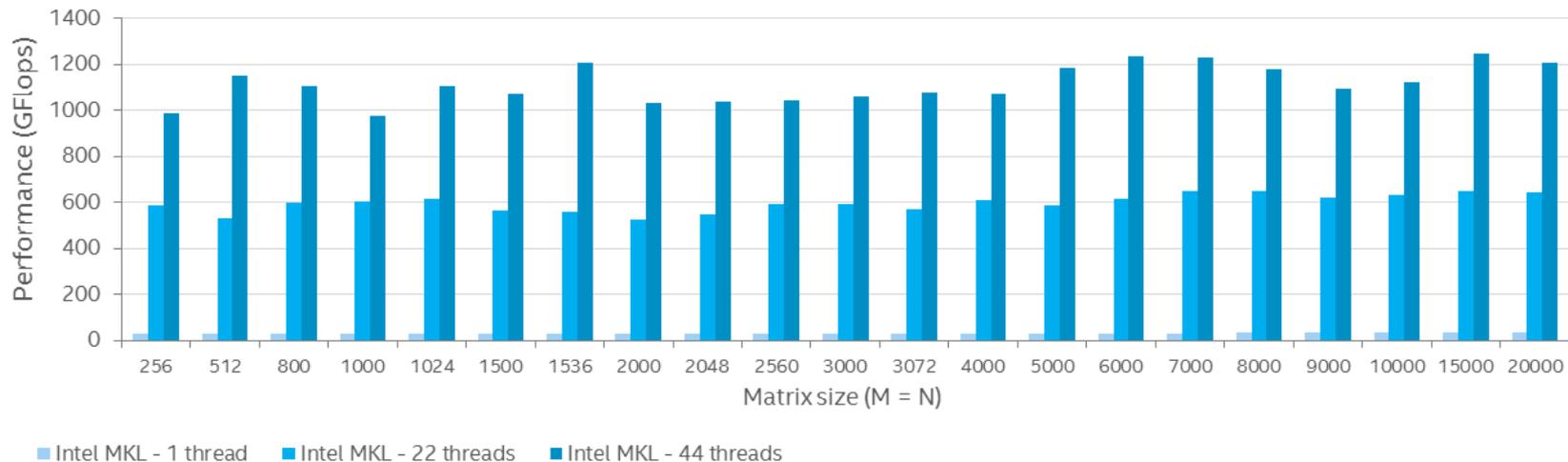
© 2016 Intel Corporation. 無断での引用、転載を禁じます。

* 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: アプリケーション・パフォーマンスの利点

DGEMM Performance

On Intel® Xeon® Processor E5-2699 v4



Configuration Info - Versions: Intel® Math Kernel Library (Intel® MKL) 2017; Hardware: Intel® Xeon® Processor E5-2699 v4, 2 Twenty-two-core CPU (55MB smart cache, 2.2GHz), 64GB of RAM; Operating System: RHEL 7.2 GA x86_64;

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. * Other brands and names are the property of their respective owners. Benchmark Source: Intel Corporation

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804.

新機能: インテル® MKL 2017

- 最適化された数学関数によりディープラーニングのニューラル・ネットワーク (CNN および DNN) に対応
- HPC クラスター上で対称固有値ソルバーの ScaLAPACK パフォーマンスを向上
- B-スプラインと単調なスプラインをベースとした新しいデータ・フィッティング関数
- インテル® Xeon Phi™ プロセッサー (開発コード名 Knights Landing) を含む最新のインテル® プロセッサー向けの最適化
- インテル® TBB のスレッド・レイヤー・サポートをすべてのレベル 1 BLAS 関数に拡張

最適化に関する注意事項

© 2016 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® DATA ANALYTICS ACCELERATION LIBRARY (インテル® DAAL)

最適化に関する注意事項

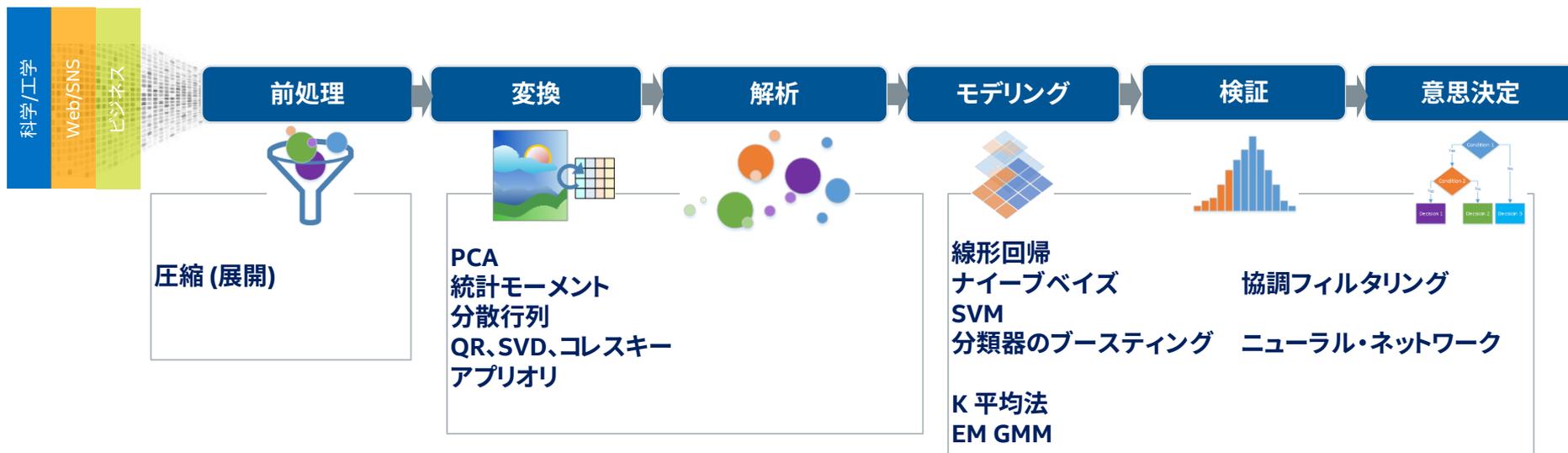
© 2016 Intel Corporation. 無断での引用、転載を禁じます。

* 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® DAAL の概要

インテル® アーキテクチャー向けに最適化されたマシンラーニングおよびディープラーニング用の最先端のパフォーマンス、C++/Java*/Python* ライブラリー



最適化に関する注意事項

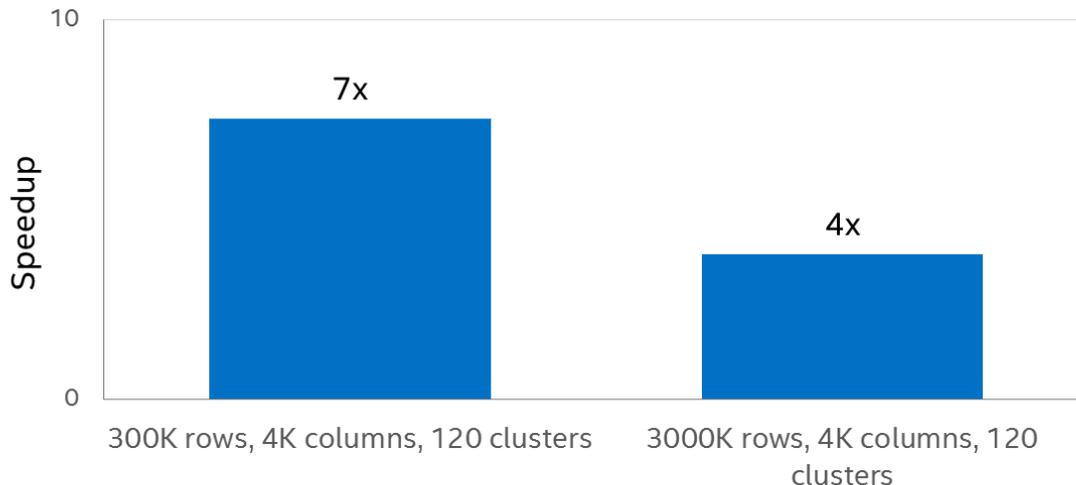
© 2016 Intel Corporation. 無断での引用、転載を禁じます。

* 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。

パフォーマンスの例: インテル® DAAL と Spark* MLLib

Intel® DAAL vs. Spark* Mllib

K-means Performance Comparison on Eight-node Cluster



Configuration Info - Versions: Intel® Data Analytics Acceleration Library 2017, Spark 1.2; Hardware: Intel® Xeon® Processor E5-2699 v3, 2 Eighteen-core CPUs (45MB LLC, 2.3GHz), 128GB of RAM per node; Operating System: CentOS 6.6 x86_64.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. * Other brands and names are the property of their respective owners. Benchmark Source: Intel Corporation

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804.

新機能: インテル® DAAL 2017

- ニューラル・ネットワーク
- Python* API (PyDAAL)
 - Anaconda または pip を利用して簡単にインストール
- KDB+ 用の新しいデータソース・コネクタ
- GitHub* のオープンソース・プロジェクト

GitHub* サイト:

<https://github.com/01org/daal> (英語)

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)

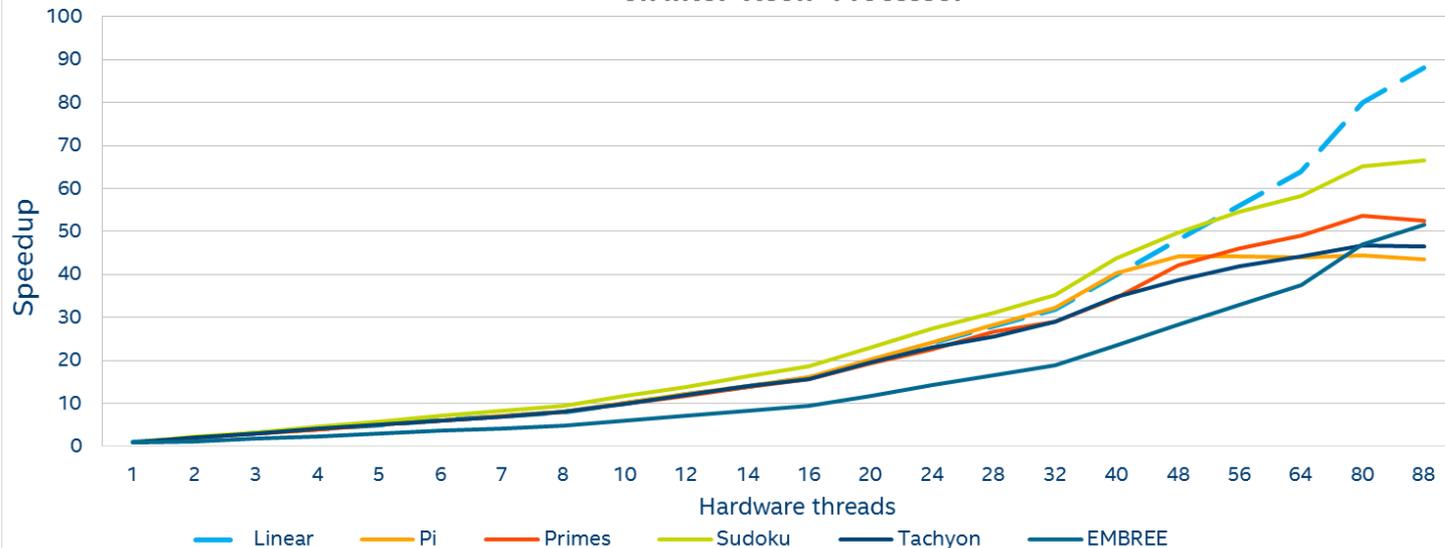
並列処理向けの豊富な機能セット

インテル® TBB

<p>汎用並列アルゴリズム</p> <p>ゼロから始めることなく、マルチコアの能力を活かす効率的でスケーラブルな方法を提供</p>	<p>フローグラフ</p> <p>並列処理を計算の依存性やデータフロー・グラフとして表すためのクラスのセット</p>	<p>コンカレント・コンテナ</p> <p>同時アクセスと、コンテナに代わるスケーラブルな手段 (外部ロックによりスレッドセーフ)</p>		
<p>タスク・スケジューラー</p> <p>並列アルゴリズムとフローグラフを強化する洗練されたワーク・スケジュール・エンジン</p>		<p>同期プリミティブ</p> <p>アトミック操作、さまざまな特性の mutex、条件変数</p>		
<p>メモリー割り当て</p> <p>スケーラブルなメモリー・マネージャーとフォルス・シェアリングのないアロケーター</p>		<p>タイマーと例外</p> <p>スレッドセーフなタイマーと例外クラス</p>	<p>スレッド</p> <p>OS API ラッパー</p>	<p>スレッド・ローカル・ストレージ</p> <p>無制限のスレッドローカル変数の効率良い実装</p>

インテル® TBB: スケーラビリティと生産性

Excellent Performance Scalability with Intel® Threading Building Blocks 2017
on Intel® Xeon® Processor



Configuration Info: Software Versions: Intel® C++ Intel® 64 Compiler, Version 17.0, Intel® Threading Building Blocks (Intel® TBB) 2017; Hardware: Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz 44/88T), 128GB Main Memory; Operating System: Red Hat Enterprise Linux Server release 7.2 (Maipo), kernel 3.10.0-327.4.5.el7.x86_64; Benchmark Source: Intel Corp. Note: sudoku, primes and tachyon are included with Intel TBB. Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, refer to www.intel.com/performance/resources/benchmark_limitations.htm.

* Other brands and names are the property of their respective owners.

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804

新機能: インテル® TBB 2017

static_partitioner クラス

- 並列ループのオーバーヘッドを最小限に抑える

streaming_node クラス

- フローグラフ内でヘテロジニアスなストリーミング計算に対応

タスクグループ/アルゴリズムの実行をスケジューラーのほかのタスクから切り分ける
メソッドの追加 (2017 のプレビュー機能)

Python* の ThreadPool クラスの代替りとなる Python* モジュールを追加

graph/stereo サンプルを追加

graph/fgbzip サンプルを改良 (async_msg の使用例を追加)

インテル® インテグレートッド・パフォーマンス・ プリミティブ (インテル® IPP)

インテル® IPP ドメインのアプリケーション

イメージ処理

- 医療用画像
- コンピューター・ビジョン
- デジタル監視
- 生体認証
- 自動ソート
- ADAS
- 視覚探索

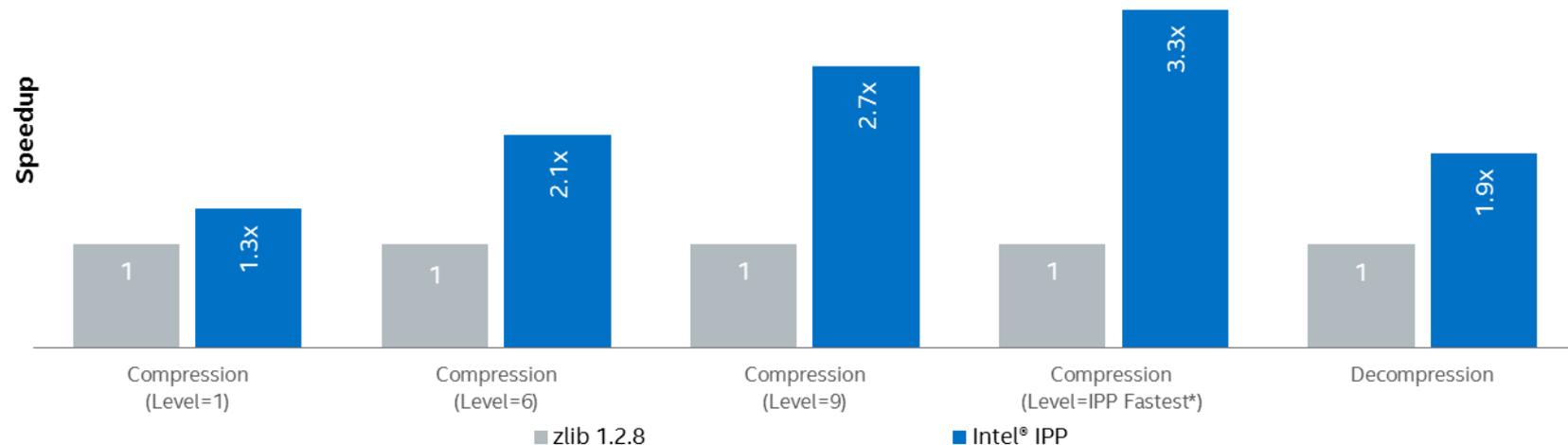
信号処理

- ゲーム (高度なオーディオコンテンツやエフェクト)
- エコー・キャンセレーション
- 通信
- エネルギー

データ圧縮と暗号化

- データセンター
- エンタープライズ・データ管理
- ID 検査
- スマートカード/
スマートウォレット
- 電子署名
- 情報セキュリティ/
サイバーセキュリティ

Intel® IPP Data Compression and Decompression Performance Boost vs. ZLIB Library



Configuration Info – SW Versions: Intel® Integrated Performance Primitives (Intel® IPP) 2017, Intel C++ Compiler 16.0. Hardware: Intel® Core™ Processor i7-6700K, 8 MB cache, 4.2 GHz, 16 GB RAM, Windows Server® 2012 R2. Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. * Other brands and names are the property of their respective owners. Benchmark Source: Intel Corporation

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804.

新機能: インテル® IPP 2017

- インテル® AVX-512、インテル® Xeon® プロセッサー、インテル® Xeon Phi™ プロセッサー/コプロセッサー向けの最適化を拡張
- 外部スレッドと 64 ビット・データをサポートするため、画像/信号処理ドメインにプラットフォーム認識 API を追加
- OpenCV* 向けのインテル® IPP の最適化機能を拡張して、zlib 圧縮関数のパフォーマンスを大幅に向上
- 次世代のインテル® Xeon Phi™ プロセッサーおよび CNL EP/XE サーバー向けの限定的なプリシリコンの最適化

インテルのソフトウェア解析ツール

インテル® VTune™ Amplifier XE - パフォーマンス・プロファイラー

インテル® Inspector - メモリー/スレッドのデバッガー

インテル® Advisor - ベクトル化の最適化とスレッドのプロトタイプ生成

インテル® VTUNE™ AMPLIFIER XE

パフォーマンス・プロファイラー

最適化に関する注意事項

© 2016 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 高速でスケーラブルなコードを迅速に開発

必要なデータを取得

- hotspot (統計コールツリー)、呼び出しカウント (統計)
- スレッド・プロファイル - コンカレンシー解析およびロックと待機の解析
- キャッシュミス、帯域幅解析¹
- GPU オフロードと OpenCL* カーネルトレース

必要な情報を素早く表示

- ソース/アセンブリで結果を表示
- OpenMP* のスケーラビリティ解析、グラフィカル・フレーム解析
- ビューポイントでデータをフィルターして関係のないデータを非表示
- スレッドおよびタスク・アクティビティをタイムライン表示

簡単に使用可能

- 特別なコンパイラーは不要 - C、C++、C#、Fortran、Java*、ASM
- Visual Studio® 統合環境またはスタンドアロン
- グラフィカル・インターフェイスとコマンドライン
- ローカルおよびリモート収集
- macOS* で Windows® および Linux* データを解析²

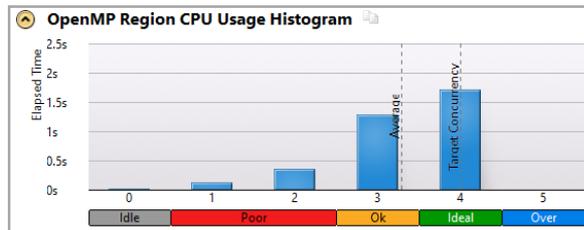
¹ プロセッサによりイベントが異なります。 ² macOS* でデータ収集はできません。

Function / Call Stack	CPU Time				Spin Time	Overhead Time
	Effective Time by Utilization	Idle	Poor	Ok		
FireObjects:checkCollision	4.507s				0s	0s
FireObjects:ProcessFireCollisionsRange	3.444s				0s	0s
NHWaitForSingleObject	0s				3.406s	0s
std::basic_istream<char,struct std::char_traits<char>>::get	3.359s				0s	0s
Ogre:FileSystemArchive:open	3.359s				0s	0s
CBaseDevice:Present	2.335s				0.671s	0s
Selected 1 row(s):					1.151s	0.728s

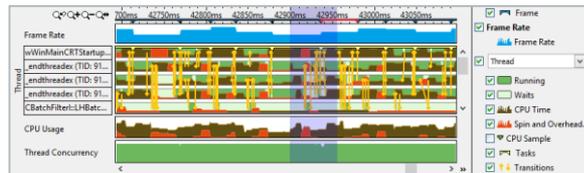
ソースコードで結果を表示

Source Line	Source	CPU Time: Total by Utilization
81	for (int i = 0; i < mem_array_i_max; i++)	0.300s
82	{	
83	for (int j = 0; j < mem_array_j_max; j++)	4.936s
84	{	
85	mem_array [j*mem_array_j_max+i] = *fill_val	7.207s

OpenMP* のスケーラビリティをチューニング



データの視覚化とフィルター



最適化に関する注意事項

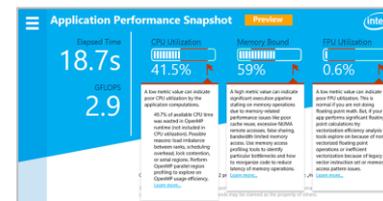
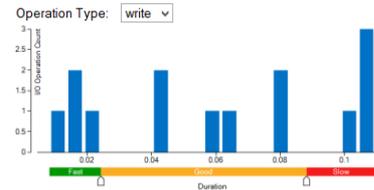
2017 の新機能: Python*、FLOPS、ストレージほか...

インテル® VTune™ Amplifier XE パフォーマンス・プロファイラー

- Python* と Python*/C++/Fortran が混在したコードのプロファイル
- 最新のインテル® Xeon Phi™ プロセッサをチューニング
- HPC パフォーマンスにとって重要な 3 つのメトリックを素早く確認
- メモリアクセスを最適化
- ストレージ解析: I/O 依存か、CPU 依存か?
- OpenCL* および GPU プロファイルの拡張
- 簡単に使用できるリモートアクセス/コマンドライン
- タイムラインにカスタムカウンターを追加可能
- プレビュー: アプリケーションとストレージのパフォーマンス・スナップショット
- インテル® Advisor: インテル® AVX-512 向けにベクトル化を最適化 (ハードウェアの有無に関係なく実行可能)



Disk Input and Output Histogram



インテル® VTune™ Amplifier XE で Knights Landing⁺ プロセッサをチューニング インテル® Xeon Phi™ プロセッサ向けの 4 つの重要な最適化

1) 高帯域メモリー

- MCDRAM に配置するデータ構造の決定
- パフォーマンスの問題をメモリー階層で表示
- DRAM および MCDRAM の帯域幅を測定

2) MPI* と OpenMP* のスケーラビリティ

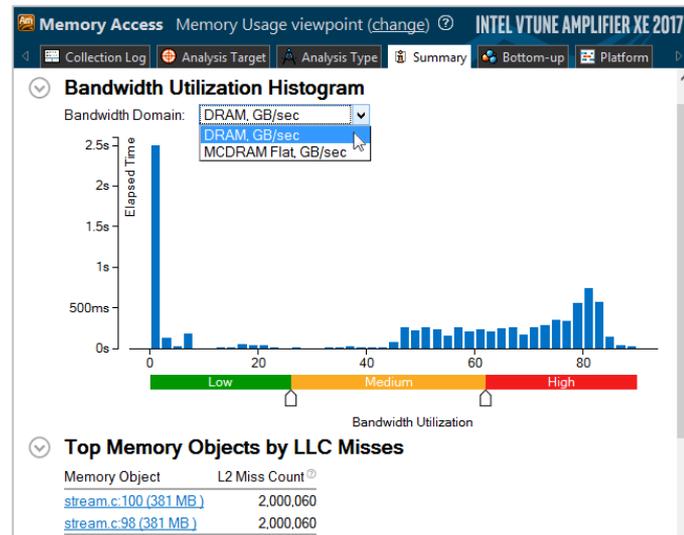
- シリアル時間と並列時間
- インバランス、オーバーヘッド・コスト、並列ループ・パラメーター

3) マイクロアーキテクチャーの効率

- コア・パイプラインにおけるコードの効率を確認
- カスタム PMU イベントで絞り込み

4) ベクトル化の効率: [インテル® Advisor](#) を使用

- インテル® AVX-512 対応ハードウェアの有無に関係なく
インテル® AVX-512 向けに最適化



† 開発コード名

最適化に関する注意事項

© 2016 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。

メモリアクセスを最適化

メモリアクセス解析: インテル® VTune™ Amplifier XE 2017

パフォーマンス向上のためデータ構造をチューニング

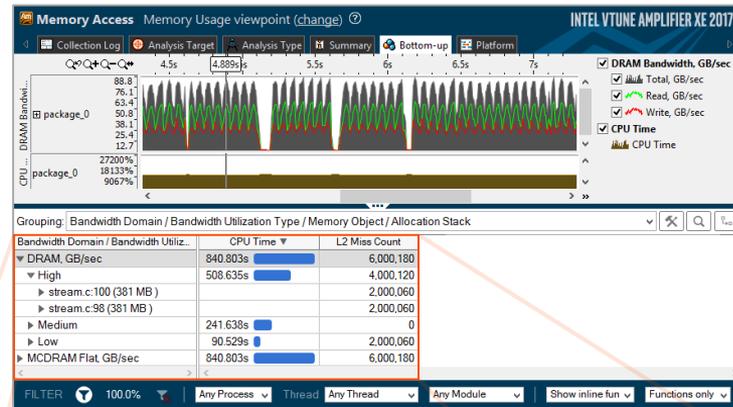
- キャッシュミス (コード行だけでなく) データ構造に紐付け
- カスタム・メモリー・アロケータのサポート

NUMAレイテンシーとスケーラビリティの最適化

- 共有とフォルス・シェアリングのチューニング
- 最大システム帯域幅を自動検出
- ソケット間の帯域幅のチューニングが容易

簡単にインストールでき、最新のプロセッサに対応

- Linux* では特別なドライバーは不要
- インテル® Xeon Phi™ プロセッサの MCDRAM (高帯域メモリー) 解析



Bandwidth Domain / Bandwidth Utiliz...	CPU Time	L2 Miss Count
▼ DRAM, GB/sec	840.803s	6,000,180
▼ High	508.635s	4,000,120
▶ stream.c:100 (381 MB)		2,000,060
▶ stream.c:98 (381 MB)		2,000,060
▶ Medium	241.638s	0
▶ Low	90.529s	2,000,060
▶ MCDRAM Flat, GB/sec	840.803s	6,000,180

最適化に関する注意事項

© 2016 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。

ストレージデバイス解析 (HDD、SATA、NVMe SSD)

インテル® VTune™ Amplifier XE

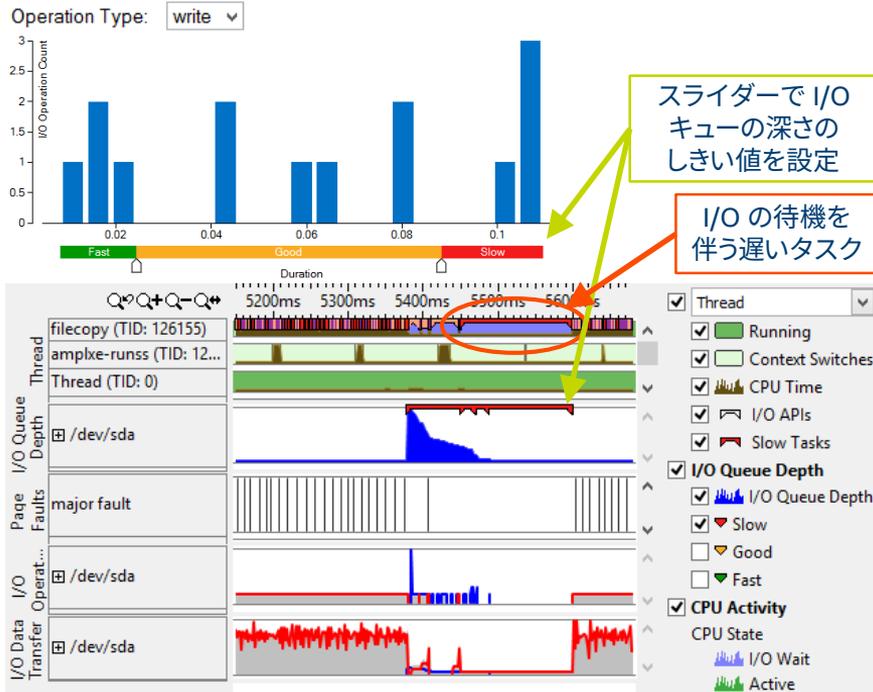
I/O 依存か、CPU 依存か?

- I/O 操作 (非同期/同期) と計算の間のインバランスを調査
- ストレージアクセスをソースコードにマップ
- CPU が I/O を待機している個所を確認
- ストレージへのバス帯域幅を測定

レイテンシー解析

- レイテンシー・ヒストグラムを利用してストレージアクセスをチューニング
- I/O を複数のデバイスに分散

Disk Input and Output Histogram



インテル® Performance Snapshots

未活用のパフォーマンスを素早く発見する 3 つの方法
アプリケーションがコンピューター・ハードウェアを有効
利用できているか？

- テストケースを実行してみてください
- ハイレベルのサマリーは、コードの現代化と高速なストレージにより利点が得られるアプリケーションを表示

パフォーマンス・スナップショットを選択:

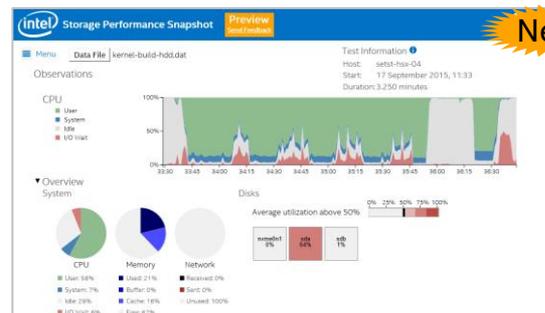
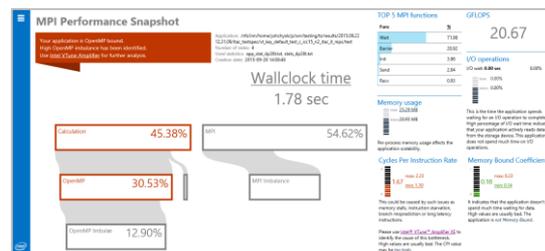
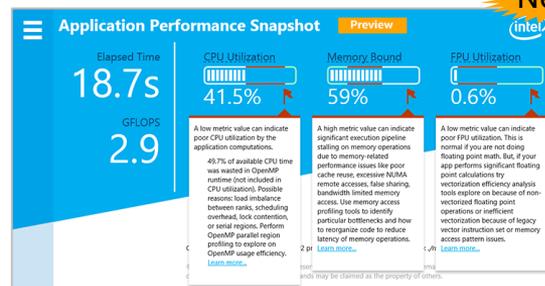
- **アプリケーション:** 非 MPI アプリケーション用
- **MPI:** MPI アプリケーション用
- **ストレージ:** ストレージが取り付けられたシステム、サーバー、ワークステーション用

無料ダウンロード: <http://www.intel.com/performance-snapshot> (英語)
インテル® Parallel Studio とインテル® VTune™ Amplifier XE にも含まれています。

最適化に関する注意事項

© 2016 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® INSPECTOR

メモリー/スレッドのデバッガー

最適化に関する注意事項

© 2016 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。

メモリー/スレッドエラーの発見とデバッグ

インテル® Inspector: メモリー/スレッドのデバッガー

正当性検証ツールにより ROI が 12%-21%¹ 向上

- 早期に問題を発見したほうが修正コストが少なく済む
- いくつかの調査 (ROI% は異なる) によると、早期に発見/対応したほうがコストを抑えられる

エラーによっては診断に数カ月を要する

- 競合やデッドロックは簡単に再現できない
- メモリーエラーをツールなしで発見するのは困難

デバッガー統合により迅速な診断が可能

- 問題の直前にブレークポイントを設定
- デバッガーで変数とスレッドを確認

数カ月かかっていた診断を数時間に短縮

¹コスト要因 - Square Project による分析

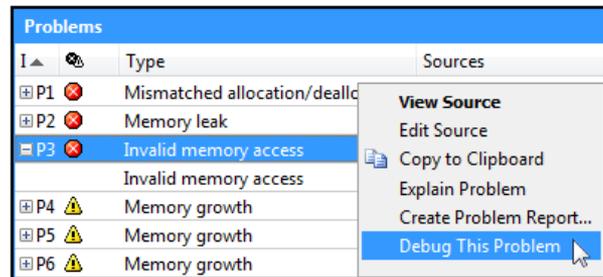
CERT: U.S. Computer Emergency Readiness Team および Carnegie Mellon CyLab
NIST: National Institute of Standards & Technology: Square Project の結果

最適化に関する注意事項

© 2016 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。

デバッガー・ブレークポイント



インテル® Parallel Studio XE Professional Edition
for Windows® および Linux* で利用可能

「インテル® Inspector により、パッケージをリリースする前に、切り分けが困難なスレッドエラーを迅速に追跡できるようになりました。」

Harmonic Inc.

ソフトウェア開発ディレクター
Peter von Kaenel 氏

<http://intel.ly/inspector-xe> (英語)



2017 の新機能: 新しいプロセッサ、新しい C++ 言語機能

インテル® Inspector 2017: メモリー/スレッドのデバッガー

新しい C++ 言語機能

- C++ 11 を完全サポート (std::mutex と std::atomic を含む)

スレッドの不具合を簡単に識別

- コード行に加えて、エラーを起こしている変数名を表示 (グローバル、スタティック、スタック変数)

インテル® Xeon Phi™ プロセッサ上でネイティブ実行

- インテル® Xeon Phi™ プロセッサ向けの開発ワークフローを単純化
- ヒント: Knights Landing[†] では、インテル® Inspector の実行中スレッド数を 30 以下にすると最良のパフォーマンスが得られる



インテル® ADVISOR

ベクトル化の最適化とスレッドのプロトタイプ生成を支援

最適化に関する注意事項

© 2016 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® Advisor により高速なコードを素早く開発 スレッドのプロトタイプ生成

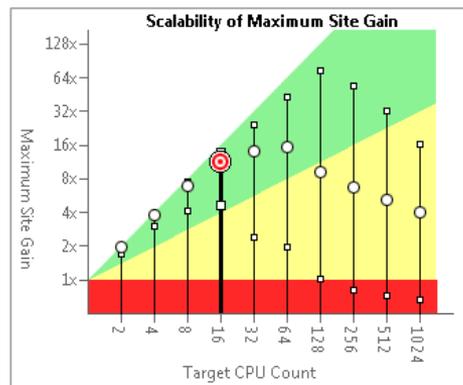
問題:

- アプリケーションをスレッド化してもパフォーマンスがそれほど向上しない
- "スケーラビリティの限界" に達したのか?
- 同期問題によりリリースを延期

データに基づくスレッド設計:

- 複数の候補のプロトタイプを素早く生成
- 大規模なシステムにおけるスケーリングを予測
- スレッド化する前に同期問題を発見
- 開発を妨げることなく設計可能

より少ない労力とリスクで、より大きな効果が
得られる並列処理を実装



「インテル® Advisor により、並列化候補の
プロトタイプを素早く生成し、開発者の時間
と労力を節約することができました。」

Sandia National Laboratories
シニア・テクニカル・スタッフ
Simon Hammond 氏

<http://intel.ly/advisor-xe> (英語)

データに基づく設計で高速なコードを素早く開発

インテル® Advisor: ベクトル化の最適化とスレッドのプロトタイプ生成

ベクトル化の最適化をスピードアップ

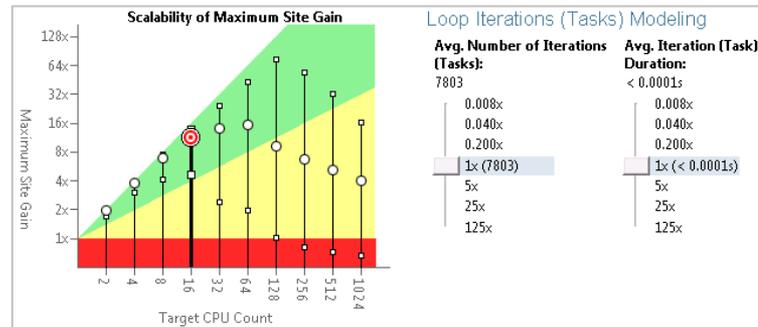
- 最も大きな利点が見られる箇所をベクトル化
- ベクトル化を妨げているものを素早く特定
- 効率良いベクトル化のためのヒント
- 安全にコンパイラによるベクトル化を強制
- メモリースタライドを最適化

スレッド設計のブレークスルー

- 複数の候補のプロトタイプを素早く生成
- 大規模なシステムにおけるスケーリングを予測
- スレッド化する前に同期問題を発見
- 開発を妨げることなく設計可能

より少ない労力とリスクで、より大きな効果

Function Call Sites and Loop	Vector Issues	Self Time	Total Time	Trip Counts	Loop Type	Why No Vectorization?	Vectorized Loops
[loop at st_algo.h:4740]		0.170s	0.170s	1	Scalar	non-vectorizable l...	
[loop at loopstl.cpp:2449]	2 Ineffective peed...	0.170s	0.170s	12; 4	Collapse	Collapse	AVX 100%
[loop at loopstl.cpp:2...]		0.150s	0.150s	12	Vectorized (B		AVX
[loop at loopstl.cpp:2...]		0.020s	0.020s	4	Remainder		
[loop at loopstl.cpp:7900]		0.170s	0.170s	500	Scalar	vectorization possi...	
[loop at loopstl.cpp:35 ...]	1 High vector regi...	0.160s	0.160s	12	Expand	Expand	AVX 68%



インテル® Parallel Studio XE for Windows® および Linux® で利用可能 <http://intel.ly/advisor-xe> (英語)

最適化に関する注意事項

© 2016 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。



2017 の新機能: インテル® AVX-512、FLOPS ほか...

インテル® Advisor: ベクトル化の最適化



- 次世代のインテル® Xeon Phi™ プロセッサをサポート
- インテル® AVX-512 対応ハードウェアの有無に関係なくインテル® AVX-512 向けのチューニングが可能
- 正確な FLOPS 計算
- メモリアクセス解析を拡張
- 影響の大きいループを簡単に選択
- バッチモードのワークフローにより時間短縮
- ループ解析により必要な情報を素早く確認

Elapsed time: 39.44s Vectorized Not Vectorized FILTER: All Modules All Sources Loops All Threads OFF Smart Mode

Summary Survey Report Refinement Reports INTEL ADVISOR 2017

Function Call Sites and Loops	Vector Issues	Self Time	Total Time	Type	FLOPS		Why No Vectorization?	Vectorized Loops				Trip Counts	Instr Trait
					GFLOPS	AI		Vect...	Efficiency	Gain...	VL (...)		
[loop in S252 at loops90.f:1172]	1 Possib...	3.080s [8.5%]	3.080s	Vectorized Ve...	0.191	0.10 ...	1 vectori...	AVX2	17%	1.38x	8	999; 62; 1	Divis
[loop in S252 at loops90.f:1172]	1 Possib..	2.970s [8.2%]	2.970s	Scalar	0.067	0.0833	vectorizat...					999	Divis
[loop in S252 at loops90.f:1172]		0.090s	0.090s	Vectorized (Bo...	4.333	0.1250		AVX2			8	62	FMA
[loop in S252 at loops90.f:1172]	1 Possib..	0.020s	0.020s	Scalar			vectorizat...					1	Divis
[loop in S2101 at loops90.f:1749]	2 Possib..	2.580s [7.1%]	2.580s	Scalar	0.152	0.0625	vectorizat...						FMA
[loop in S126 at loops90.f:447]	1 Assu ...	1.068s	1.068s	Scalar	0.370	0.1667	vector de ...						
[loop in S343 at loops90.f:2300]	1 Assu ...	1.020s	1.020s	Scalar			vector de ...						
[loop in S353 at loops90.f:2381]	1 Possib..	0.880s	0.880s	Vectorized (Bo...	2.274	0.1250		AVX2	35%	2.78x	8	62; 4; 1	FMA

最適化に関する注意事項

© 2016 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。

クラスターツール

インテル® MPI ライブラリー

インテル® Trace Analyzer & Collector

インテル® MPI ライブラリーの概要

最適化された MPI アプリケーション・パフォーマンス

- アプリケーション固有のチューニング
- 自動チューニング
- New! - インテル® Xeon Phi™ プロセッサ (開発コード名 Knights Landing) をサポート
- New! - インテル® Omni-Path アーキテクチャー・ベースのファブリックをサポート

低レイテンシーおよび複数のベンダーとの互換性

- 業界トップレベルのレイテンシー
- OpenFabrics* インターフェイス (OFI) により、ファブリック向けに最適化されたパフォーマンスをサポート

高速な MPI 通信

- 最適化された集合操作

持続性のあるスケーラビリティ (最大 34 万コアまで)

- ネイティブ InfiniBand* インターフェイス・サポートにより、低レイテンシー、高帯域幅、メモリー使用量の軽減を実現

安定性に優れた MPI アプリケーション

- インテル® Trace Analyzer & Collector とシームレスに連携



最適化に関する注意事項

© 2016 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。

新機能: インテル® MPI ライブラリー 2017

- インテル® Xeon Phi™ プロセッサ (開発コード名 Knights Landing) をサポート
- インテル® Omni-Path アーキテクチャー・ベースのファブリックをサポート
- KNL⁺ 向けに最適化された memcpy の使用
- 1 つの KNL⁺ ノードに対する共有メモリー集合操作のチューニング
- RMA の一般的な最適化
- 一般的な最適化、起動時間の短縮、MPI チューニング・ユーティリティーの高速化

⁺ 開発コード名 Knights Landing の略称

最適化に関する注意事項

© 2016 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。



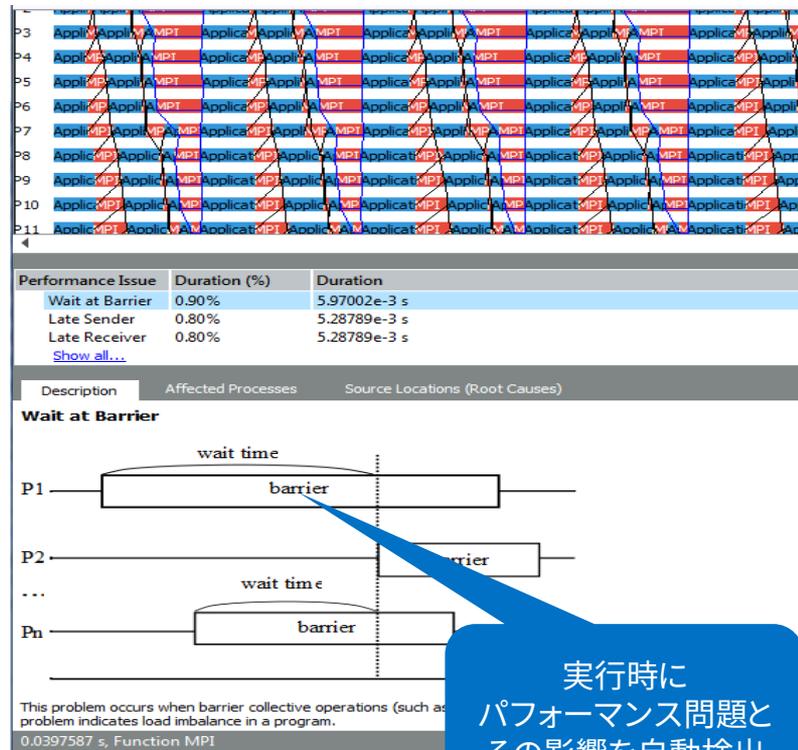
インテル® Trace Analyzer & Collector の概要

開発者を支援

- 並列アプリケーションの動作を視覚化して確認
- プロファイル統計とロードバランスを評価
- 通信 hotspot を特定

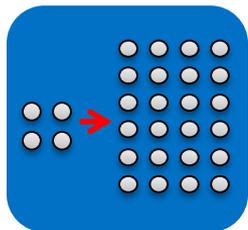
機能

- イベントベースのアプローチ
- 低オーバーヘッド
- 優れたスケーラビリティ
- 強力な集合およびフィルター関数
- アイデアライザー

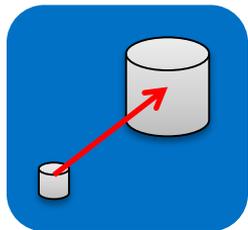


MPI* Performance Snapshot

MPIとハイブリッドのスケラブルなプロファイル



軽量: 100K ランクを低オーバーヘッドでプロファイル

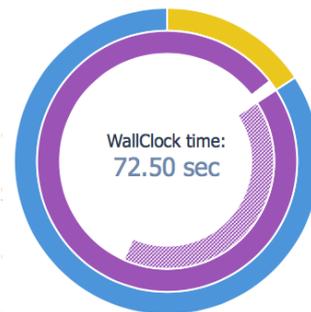


スケラブル: スケーリングによるパフォーマンスの変化を迅速に検出



主要メトリック: MPI/OpenMP*のインバランスを表示

MPI Performance Snapshot Summary



■ WallClock time: 72.50 sec
Total application lifetime. The time is elapsed time for the slowest process. This metric includes the MPI Time and the Computation time below.

■ MPI Time: 11.46 sec 15.81%
Time spent inside the MPI library. High values are usually bad. This value is **AVERAGE**. The application is **Communication-bound**. [More details...](#)

⚡ MPI Imbalance: 9.32 sec 12.86%
Mean unproductive wait time per process spent in the MPI library calls when a process is waiting for data. This time is part of the MPI time above. High values are usually bad. This value is **AVERAGE**. The application workload is **NOT well balanced** between MPI ranks. [More details...](#)

■ Computation Time: 61.03 sec 84.19%
Mean time per process spent in the application code. This is the sum of the OpenMP Time and the Serial time. High values are usually good. This value is **HIGH**. The application is **Computation-bound**. [More details...](#)

■ OpenMP Time: 71.54 sec 98.68%
Mean time per process spent in the OpenMP parallel regions. High values are usually good and indicate that the application is well-threaded. This value is **HIGH**.

⚡ OpenMP Imbalance: 29.59 sec 40.82%
Mean unproductive wait time per process spent in OpenMP parallel regions (normally at synchronization barriers). High values are usually bad. This value is **HIGH**. The application's OpenMP work sharing is **NOT well load-balanced**. [More details...](#)

■ Serial Time: 0.00 sec 0.00%
Mean application time per process spent outside OpenMP parallel regions. High values may be good or bad depending on the application algorithm. This value is **NEGLIGIBLE**. This application is **well parallelized** via OpenMP directives.

新機能: インテル® Trace Analyzer & Collector

- 開発コード名 Knights Landing に対応予定
- インバランス・プロファイラーのスケールビリティが最大 10 倍向上
- MPI Performance Snapshot 機能の HTML 出力が向上

関連情報 (英語)

- [製品ページ](#) – 概要、機能、FAQ、サポート...
- [トレーニング資料](#) – 動画、技術資料、ドキュメント...
- [評価ガイド](#) – 基本的な操作手順
- [お客様の声](#)

その他の開発製品:

- [インテル® ソフトウェア開発製品](#)

BACK-UP

Enhanced Application Performance with Intel® AVX-512 Support

Enhanced performance due to Intel® AVX-512 instructions taking advantage of FMA units, memcopy, new pre-fetch instructions, new transcendental instructions, MCDRAM, and increased number of cores.

Enhanced Application Performance with AVX-512 Support

	Key functionality / library domain	KNL features used to deliver enhanced performance (instructions, other)	
Intel® Math Kernel Library	*GEMMs/BLAS MP Linpack	<ul style="list-style-type: none"> Two FMA units + 2 instruction decoders are key AVX512 FMA (vfmadd231ps or vfm231pd) 	<ul style="list-style-type: none"> Prefetcht0 instruction MCDRAM
	LU/Cholesky/QR/LAPACK/SMP Linpack	Same as in BLAS (as main LAPACK kernel is ?*GEMM) + greater core count	
	2D and 3D FFTs	<ul style="list-style-type: none"> Two FMA units + 2 instruction decoders MCDRAM, tile-to-tile mesh 	<ul style="list-style-type: none"> AVX512 FMA Prefetcht1 instruction
	DNN	<ul style="list-style-type: none"> Two FMA units + 2 instruction decoders MCDRAM, tile-to-tile mesh AVX512 FMA 	<ul style="list-style-type: none"> Prefetcht0, prefetcht1 instruction Masking support Large core count
	Sparse	<ul style="list-style-type: none"> Two FMA units + 2 instruction decoders MCDRAM AVX512 FMA 	<ul style="list-style-type: none"> Prefetcht1 instruction Depend on seq. Blas level 3 Knights Landing improvement
	Vector Statistics	Similar to BLAS/LAPACK, greater number of cores	
	Vector Math	<ul style="list-style-type: none"> AVX512 FMA Two FMA units + 2 instruction decoders Large number of cores for MT performance 	<ul style="list-style-type: none"> New Transcendental Support Instructions: VGETEXP, VGETMANT, VRNDSCALE, VSCALEF, VFIXUPIMM, VRCP28, VRSQRT28, VEXP2
Intel® Integrated Performance Primitives	All – from Signal Processing (1D) and up to Image (2D) and Volume (3D) processing	The main advantage inherited from LRB/KNL is support of mask registers and therefore support of predicates for all new instructions. Then, - full 512-bit register palign support (no lanes restrictions as for "old" AVX palign)- <code>_mm512_alignr_epi32</code> , <code>_mm512_alignr_epi64</code> . Then, on the "fly" integer conversions: <code>vpmovq{w b d}</code> , <code>vpmovq{w b}</code> . And the last one – integer any-direction comparison: <code>vpcmp{d q}</code> and <code>vpcmpu{d q}</code> .	
Intel® Data Analytics Acceleration Library		Similar to BLAS/LAPACK, greater number of cores	
Intel® MPI Library		<ul style="list-style-type: none"> Used compiler's AVX-512 version of memcpy (but w/ fix, failed CQ on ICC) Build IMPI w/ -fvisibility=hidden (make all symbols as hidden by default and only needed as external). Addressed KNL micro-arch features, such as short BTB, by reducing access to PLT/GOT Reduced/simplified critical path where it's possible. Addressed KNL frond-end specifics. 	

最適化に関する注意事項

© 2016 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。

Easy Access to Intel® Parallel Studio XE Runtimes

For Amazon Web Services* users only

Intel Parallel Studio XE Runtime

- Required to be able to run applications built with the Intel® Performance Libraries or Intel® compilers.
- Includes latest optimizations for Intel® architecture for faster application performance
- Linux* only

Easy access for Amazon Web Services users at no cost

- Latest runtimes through Linux native repos
 - YUM repo – available now! (<http://bit.ly/ParallelStudioXE-Runtimes>)

Educating with Webinar Series about 2017 Tools

- Expert talks about the new features
- Series of live webinars, September 13 – November 8, 2016
- Attend live or watch after the fact.

<https://software.intel.com/events/hpc-webinars>

What's New In Intel® Parallel Studio XE 2017 (Online)

Sep 13, 2016 (9:00am - 10:00am PST)

This webinar will go over the latest features of the new release of Intel® Parallel Studio XE 2017.



Add

Parallel Programming and Optimization for Intel® Architecture (Workshop)

São Paulo, Brazil

Sep 21, 2016 (11:00am - 5:00pm EBT)

Offered by UNESP in partnership with Intel software Brazil, aims to a practical approach to parallel programming on Intel® Xeon® and Intel® Xeon Phi™ based systems



Add

Code for Speed with High Bandwidth Memory on Intel® Xeon Phi™ Processors (Online)

Oct 11, 2016 (9:00am - 10:00am PST)

Cover methods for users to analyze suitable memory mode and "memkind" library interface, a user-extensible heap manager built on top of jemalloc.



Add

Vectorization, the "Other" Parallelism You Need (Online)

Oct 18, 2016 (9:00am - 10:00am PST)

This session demonstrates how process of identifying and modifying code to take advantage of the vector hardware will boost application performance.



Add

Roofline analysis: A new way to visualize performance optimization tradeoffs (Online)

Oct 25, 2016 (9:00am - 10:00am PST)

Join us in the webinar to see a demonstration and an introduction to how to use roofline analysis to make your own code more efficient.



Add

最適化に関する注意事項

© 2016 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。

Educating with High-Performance Programming Book

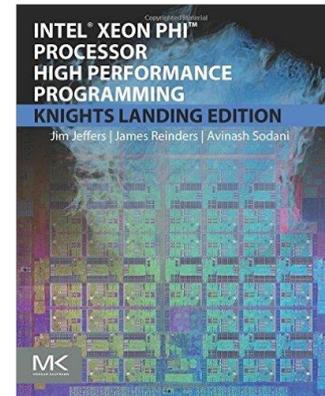
Knights-Landing-specific details, programming advice, and real-world examples.

Intel® Xeon Phi™ Processor High Performance Programming

- Techniques to generally increase program performance on any system and prepare you better for Intel Xeon Phi processors.

Available as of June 2016

<http://lotsofcores.com>



"I believe you will find this book is an invaluable reference to help develop your own Unfair Advantage."

James A.
Manager
Sandia National Laboratories

More Education with software.intel.com/moderncode

- Online community growing collection of tools, trainings, support
 - Features Black Belts in parallelism from Intel and the industry
- Intel® HPC Developer Conferences developers share proven techniques and best practices
 - hpcdevcon.intel.com
- Hands-on training for developers and partners with remote access to Intel® Xeon® processor and Xeon Phi™ coprocessor-based clusters.
 - software.intel.com/icmp
- Developer Access Program provides early access to Intel® Xeon Phi™ processor codenamed Knights Landing plus one-year license for Intel® Parallel Studio XE Cluster Edition.
 - <http://dap.xeonphi.com/>



Choices to Fit Needs: Intel® Tools

All Products with support – worldwide, for purchase.

- Intel® Premier Support - private direct support from Intel
- support for past versions
- software.intel.com/products

Most Products without Premier support – via special programs for those who qualify

- students, educators, classroom use, open source developers, and academic researchers
- software.intel.com/qualify-for-free-software
- Intel® Performance Libraries without Premier support -Community licensing for Intel performance libraries
 - no royalties, no restrictions based on company or project size
 - software.intel.com/nest

Free Software Tools
Supporting qualified students, educators, academic researchers and open source contributors

Free Intel® Software Development Tools for:

- Academic Researcher
For academic research at institutions of higher education.
- Student
For current students at degree-granting institutions.
- Educator
For use in teaching curriculum.
- Open Source Contributor
For developers actively contributing to open source projects.

Community support only – all tools:
Students, Educators, classroom use,
Open Source Developers,
Academic Researchers (qualification required)

Intel
TAKE FLIGHT
LET YOUR CODE SOAR

WELCOME TO THE AVIARY.
Learn more here about your feathered friends.

Community support only – Intel Performance Libraries:
Community Licensing (no qualification required)

最適化に関する注意事項

© 2016 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。

What's New: Details

Intel® C++ Compiler

- SIMD Data Layout Templates to facilitate vectorization for your C++ code
- Virtual function vectorization capability
- Enhanced C11 and C++14 language standards support
 - Sized deallocation
 - Relaxed constexpr restrictions
 - Variable templates
 - Single-Quotation-Mark as a digit separator,
- Enhanced GNU* and Microsoft* compatibility
- SSE Cast Support
- Diagnostic improvements on template argument
- Support for a range of target operating systems, including Android* and embedded Linux OS's

What's New: Details

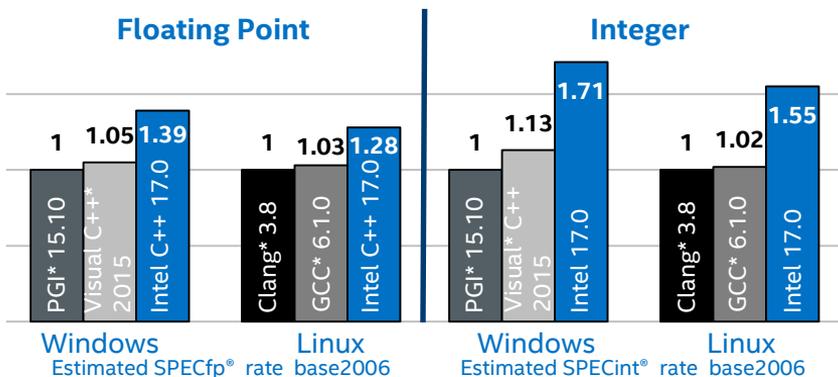
Intel® Fortran Compiler

- Substantial Coarray Fortran* performance improvement on non-trivial programs
- Almost complete Fortran 2008 support
- Enhanced Fortran 2008 and draft Fortran 2015 language standards support
 - implied-shape PARAMETER arrays
 - 2008 bind C internal procedures
 - extended EXIT for all named blocks
 - pointer initialization
- VS2013 Shell* replaces VS2010 Shell on Windows*

Intel® C++ Compilers

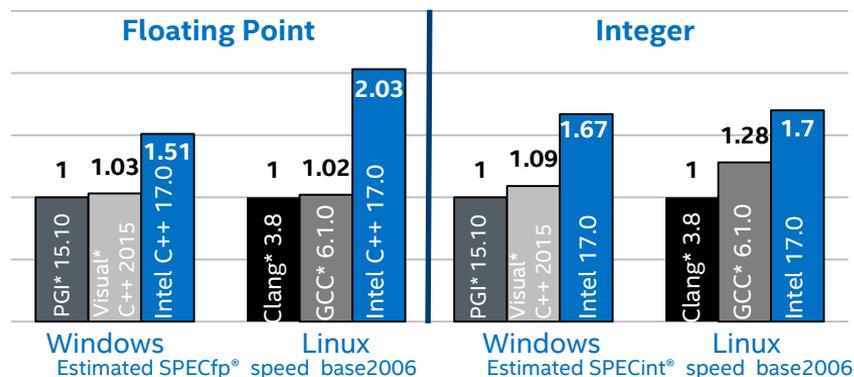
Performance Advantage as Measured by SPEC*

Boost C++ application performance
on Windows* & Linux* using Intel® C++ Compiler
(higher is better)



Relative geomean performance, SPEC* benchmark - higher is better

Boost C++ application performance
on Windows* & Linux* using Intel® C++ Compiler
(higher is better)



Relative geomean performance, SPEC* benchmark - higher is better

Configuration: Windows hardware: Intel(R) Xeon(R) CPU E3-1245 v5 @ 3.50GHz, HT enabled, TB enabled, 32 GB RAM; Linux hardware: Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz, 256 GB RAM, HyperThreading is on.
Software: Intel compilers 17.0, Microsoft (R) C/C++ Optimizing Compiler Version 19.00.23918 for x86/x64, GCC 6.1.0, PGI 15.10, Clang/LLVM 3.8, Linux OS: Red Hat Enterprise Linux Server release 7.1 (Maipo), kernel 3.10.0-229.el7.x86_64, Windows OS: Windows 10 Pro (10.0.10240 N/A Build 10240).
SPEC* Benchmark (www.spc.org). SmartHeap lib 11.3 for Visual C++ and Intel Compiler were used for SPECint* benchmarks.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. * Other brands and names are the property of their respective owners. Benchmark Source: Intel Corporation

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804.

Configuration: Windows hardware: Intel(R) Xeon(R) CPU E3-1245 v5 @ 3.50GHz, HT enabled, TB enabled, 32 GB RAM; Linux hardware: Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz, 256 GB RAM, HyperThreading is on.
Software: Intel compilers 17.0, Microsoft (R) C/C++ Optimizing Compiler Version 19.00.23918 for x86/x64, GCC 6.1.0, PGI 15.10, Clang/LLVM 3.8, Linux OS: Red Hat Enterprise Linux Server release 7.1 (Maipo), kernel 3.10.0-229.el7.x86_64, Windows OS: Windows 10 Pro (10.0.10240 N/A Build 10240).
SPEC* Benchmark (www.spc.org).

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. * Other brands and names are the property of their respective owners. Benchmark Source: Intel Corporation

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804.

最適化に関する注意事項

© 2016 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。



Impressive Performance Improvement

Intel® Compiler OpenMP* Explicit Vectorization

- Three lines added that take full advantage of both SSE or AVX
- Pragma's ignored by other compilers so code is portable

```
#pragma omp declare simd linear(z:40) uniform(L, N, Nmat) linear(k)
float path_calc(float *z, float L[][VLEN], int k, int N, int Nmat)
```

```
#pragma omp declare simd uniform(L, N, Nopt, Nmat) linear(k)
float portfolio(float L[][VLEN], int k, int N, int Nopt, int Nmat)
```

```
... ..
for (path=0; path<NPATH; path+=VLEN) {
    /* Initialise forward rates */
    z = z0 + path * Nmat;
```

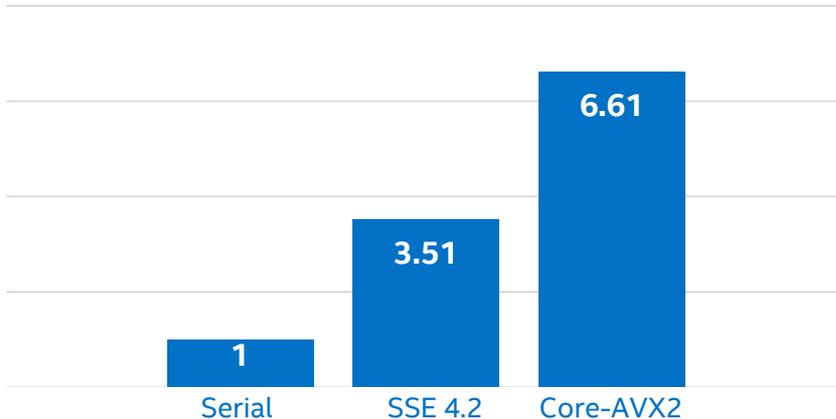
```
#pragma omp simd linear(z:Nmat)
for(int k=0; k < VLEN; k++){
    for(i=0;i<N;i++){
        L[i][k] = LO[i];
    }
}
```

```
/* LIBOR path calculation */
float temp = path_calc(z, L, k, N, Nmat);
v[k+path] = portfolio(L, k, N, Nopt, Nmat);
```

```
/* move pointer to start of next block */
z += Nmat;
}
}
```

Libor calculation speedup

Normalized performance data – higher is better



Configuration: Intel® Xeon® CPU E3-1270 @ 3.50 GHz Haswell system (4 cores with Hyper-Threading On), running at 3.50GHz, with 32.0GB RAM, L1 Cache 256KB, L2 Cache 1.0MB, L3 Cache 8.0MB, 64-bit Windows® Server 2012 R2 Datacenter. Compiler options: SSE4.2: -O3 -Qopenmp-simd -QxSSE4.2 or AVX2: -O3 -Qopenmp-simd -QxCORE-AVX2. For more information go to <http://www.intel.com/performance>

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. * Other brands and names are the property of their respective owners. Benchmark. Source: Intel Corporation

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSE3.5 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804.

最適化に関する注意事項

© 2016 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。

Impressive Performance Improvement

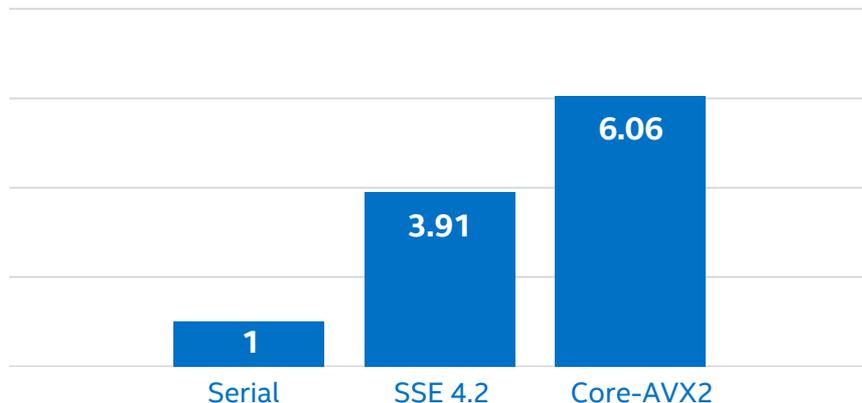
Intel C++ Explicit Vectorization: SIMD Performance

- One line added that take full advantage of both SSE or AVX
- Pragma's ignored by other compilers so code is portable

```
#pragma simd vectorlength(8)
for (int x = x0; x < x1; ++x) {
    float div = coef[0] * A_cur[x]
        + coef[1] * ((A_cur[x + 1] + A_cur[x - 1])
        + (A_cur[x + Nx] + A_cur[x - Nx])
        + (A_cur[x + Nxy] + A_cur[x - Nxy]))
        + coef[2] * ((A_cur[x + 2] + A_cur[x - 2])
        + (A_cur[x + sx2] + A_cur[x - sx2])
        + (A_cur[x + sxy2] + A_cur[x - sxy2]))
        + coef[3] * ((A_cur[x + 3] + A_cur[x - 3])
        + (A_cur[x + sx3] + A_cur[x - sx3])
        + (A_cur[x + sxy3] + A_cur[x - sxy3]))
        + coef[4] * ((A_cur[x + 4] + A_cur[x - 4])
        + (A_cur[x + sx4] + A_cur[x - sx4])
        + (A_cur[x + sxy4] + A_cur[x - sxy4]));
    A_next[x] = 2 * A_cur[x] - A_next[x] + vsq[s+x] * div;
}
```

RTM-stencil calculation speedup

Normalized performance data – higher is better



Configuration: Intel® Xeon® CPU E3-1270 @ 3.50 GHz Haswell system (4 cores with Hyper-Threading On), running at 3.50GHz, with 32.0GB RAM, L1 Cache 256KB, L2 Cache 1.0MB, L3 Cache 8.0MB, 64-bit Windows® Server 2012 R2 Datacenter. Compiler options: SSE4.2: -O3 -Qopenmp -simd -QxSSE4.2 or AVX2: -O3 -Qopenmp -simd -QxCORE-AVX2. For more information go to <http://www.intel.com/performance>

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. * Other brands and names are the property of their respective owners. Benchmark Source: Intel Corporation

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804.

最適化に関する注意事項

© 2016 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。

SIMD Data Layout Template

Improve Productivity and Boost C++ Performance

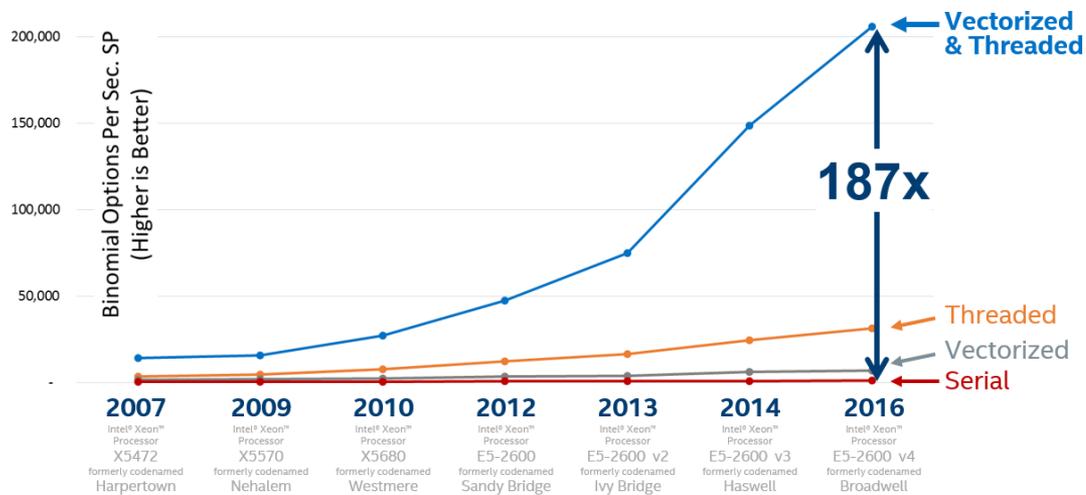
- Quickly convert “Array of Structures” to “Structure of Arrays” representation.
- Increase productivity: Use predefined templates with minimal effort, and let SDLT do the vectorization for you.
- Improve performance: SDLT vectorizes your code by making memory access contiguous, which can lead to more efficient code and better performance.
- Seamless integration: SDLT follows the familiar Intel vector programming model.

“We used SDLT to vectorize the deformer code in Premo, the in-house animation tool for DreamWorks Animation. The performance improvements we were able to achieve were dramatic, and these improvements will translate directly into higher quality characters that will be seen on-screen in future movies. Also the library itself was easy to use and integrate into our existing codebase.”

Martin Watt
Principal Engineer,
DreamWorks Animation

Intel® Advisor: Modernize Your Code

Vectorization Optimization and Thread Prototyping

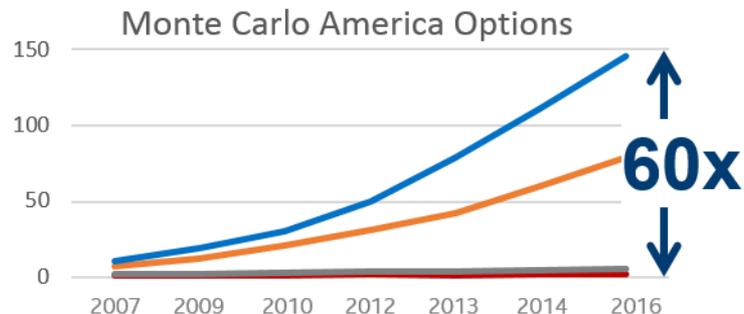
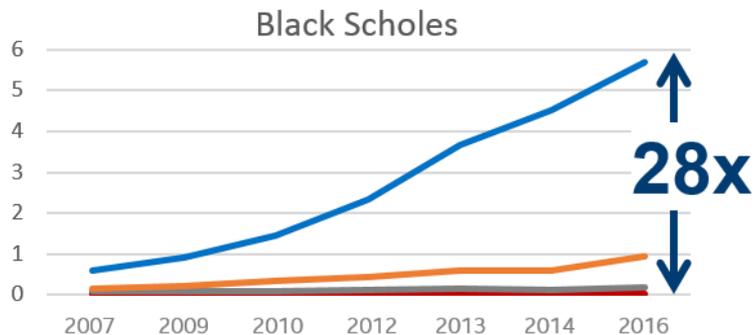
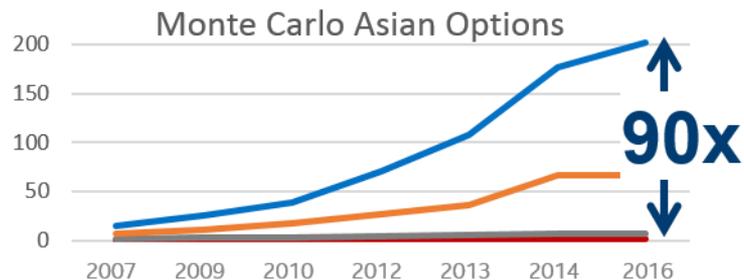
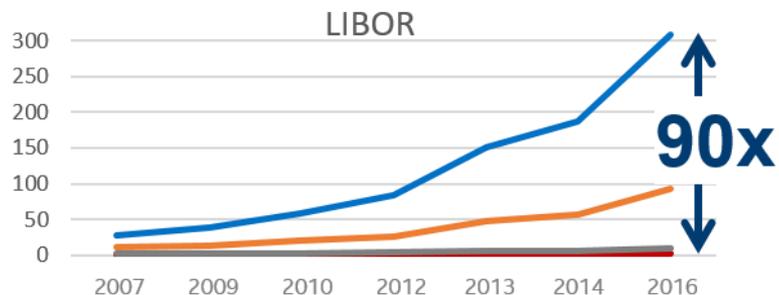


The Difference Is Growing With Each New Generation of Hardware

- Vectorize and thread your code or performance “dies” on modern processors
- Get trip counts, data dependencies, memory access patterns, and more
- Follow an easy optimization workflow with tips for faster code

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to <http://www.intel.com/performance> [Configurations](#) at the end of this presentation.

Vectorization and Threading Critical on Modern Hardware



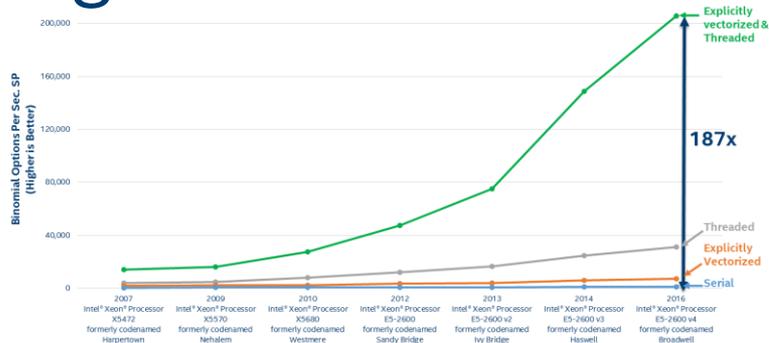
Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to <http://www.intel.com/performance> [Configurations](#) at the end of this presentation.

最適化に関する注意事項

© 2016 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。

Configurations for Binomial Options SP



Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804

Performance measured in Intel Labs by Intel employees

Platform Hardware and Software Configuration

Platform	Unscaled Core Frequency	Cores/Socket	Num Sockets	L1 Data Cache	L1 I Cache	L2 Cache	L3 Cache	Memory	Memory Frequency	Memory Access	H/W Prefetchers Enabled	HT Enabled	Turbo Enabled	C States	O/S Name	Operating System	Compiler Version
Intel® Xeon™ 5472 Processor	3.0 GHZ	4	2	32K	32K	12 MB	None	32 GB	800 MHZ	UMA	Y	N	N	Disabled	Fedora 20	3.11.10-301.fc20	icc version 14.0.1
Intel® Xeon™ X5570 Processor	2.93 GHZ	4	2	32K	32K	256K	8 MB	48 GB	1333 MHZ	NUMA	Y	Y	Y	Disabled	Fedora 20	3.11.10-301.fc20	icc version 14.0.1
Intel® Xeon™ X5680 Processor	3.33 GHZ	6	2	32K	32K	256K	12 MB	48 MB	1333 MHZ	NUMA	Y	Y	Y	Disabled	Fedora 20	3.11.10-301.fc20	icc version 14.0.1
Intel® Xeon™ E5 2690 Processor	2.9 GHZ	8	2	32K	32K	256K	20 MB	64 GB	1600 MHZ	NUMA	Y	Y	Y	Disabled	Fedora 20	3.11.10-301.fc20	icc version 14.0.1
Intel® Xeon™ E5 2697v2 Processor	2.7 GHZ	12	2	32K	32K	256K	30 MB	64 GB	1867 MHZ	NUMA	Y	Y	Y	Disabled	Fedora 20	3.11.10-301.fc20	icc version 14.0.1
Intel® Xeon™ E5 26xxv3 Processor	2.2 GHZ	14	2	32K	32K	256K	35 MB	64 GB	2133 MHZ	NUMA	Y	Y	Y	Disabled	Fedora 20	3.13.5-202.fc20	icc version 14.0.1
Intel® Xeon™ E5 26xxv4 Processor																	

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to <http://www.intel.com/performance>

最適化に関する注意事項

© 2016 Intel Corporation. 無断での引用、転載を禁じます。

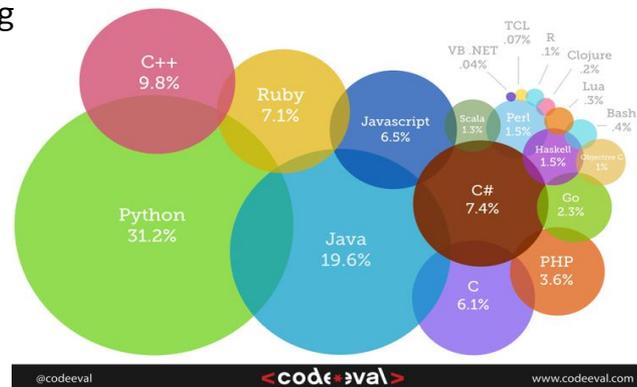
* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。



Python* Landscape

Adoption of Python continues to grow among domain specialists and developers for its productivity benefits

Most Popular Coding Languages of 2015

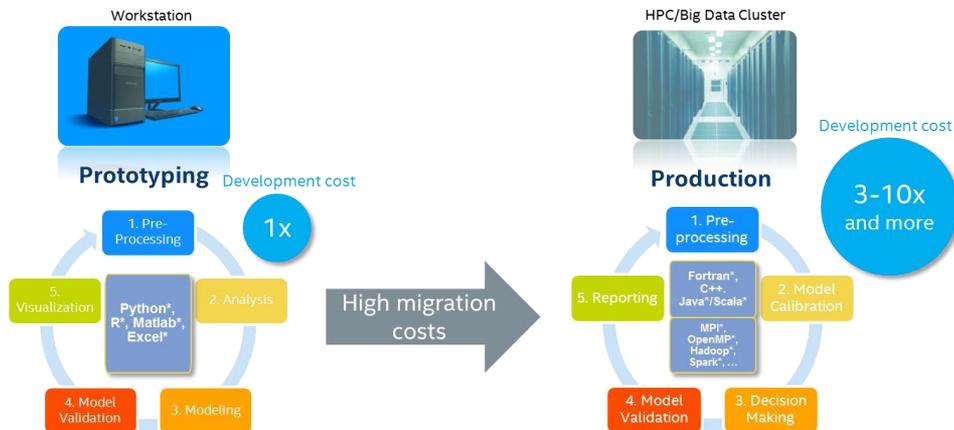


Challenge#1:

Domain specialists are not professional software programmers.

Challenge#2:

Python performance limits migration to production systems



最適化に関する注意事項

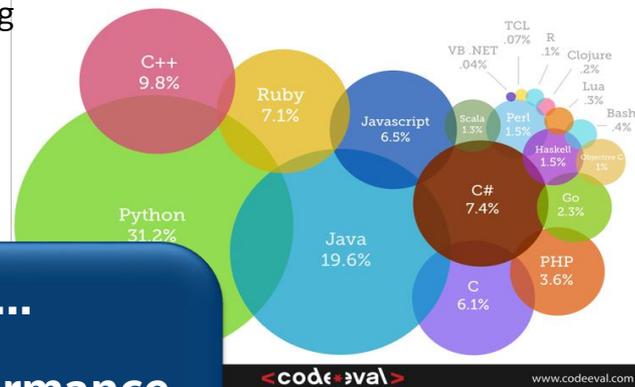
© 2016 Intel Corporation. 無断での引用、転載を禁じます。

* 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。

Python* Landscape

Adoption of Python continues to grow among domain specialists and developers for its productivity benefits

Most Popular Coding Languages of 2015



Challenge#1:

Domain specialists are not professional programmers.

Challenge#2:

Python performance limits migration to production systems

Intel's solution is to...

- Accelerate Python performance
- Enable easy access
- Empower the community



High migration costs



最適化に関する注意事項

© 2016 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。

Access Multiple Options for Faster Python*

Included in Intel® Distribution for Python

Accelerate with native libraries

- NumPy, SciPy, Scikit-Learn, Theano, Pandas, pyDAAL
- Intel® MKL, Intel® DAAL

Exploit vectorization and threading

- Cython + Intel C++ compiler
- Numba + Intel LLVM

Better/Composable threading

- Cython, Numba, Pyston
- Threading composability for MKL, CPython, Blaze/Dask, Numba

"I expected Intel's numpy to be fast but it is significant that plain old python code is much faster with the Intel version too."

 Puget
systems

Dr. Donald Kinghorn,
Puget Systems [Review](#)

Multi-node parallelism

- Mpi4Py, Distarray
- Intel native libraries: Intel MPI

Integration with Big Data, ML platforms and frameworks

- Spark, Hadoop, Trusted Analytics Platform

 Work in Progress

Better performance profiling

- Extensions for profiling mixed Python & native/JIT codes

Intel® Distribution for Python* Reviews

InfoWorld

**Intel's Python
distribution provides a
major math boost**

The still-in-beta Python distribution uses Math Kernel Library to speed up processing on Intel hardware

The distribution's main touted advantage is speed -- but not a PyPy-style general speedup via a JIT. Instead, the MKL speeds up certain math operations so that they run faster on one thread and multiple threads.

"I expected Intel's numpy to be fast but it is significant that plain old python code is much faster with the Intel version too."

Puget
systems

Dr. Donald Kinghorn,
Puget Systems [Review](#)

 insideHPC

**HPC Podcast Looks at
Intel's Pending
Distribution of
Python**

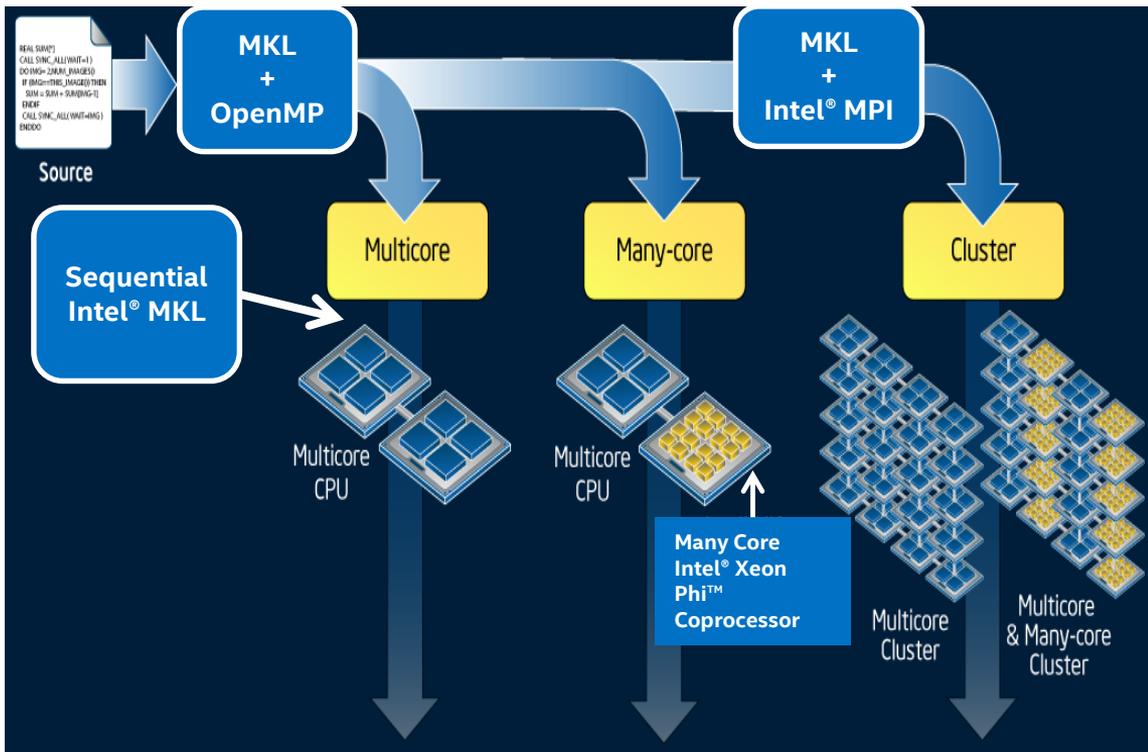
Yes, Intel is doing their own Python build! It is still in beta but I think it's a great idea.Yeah, it's important!

Automatic Performance Scaling from the Core, to Multicore, to Many Core and Beyond

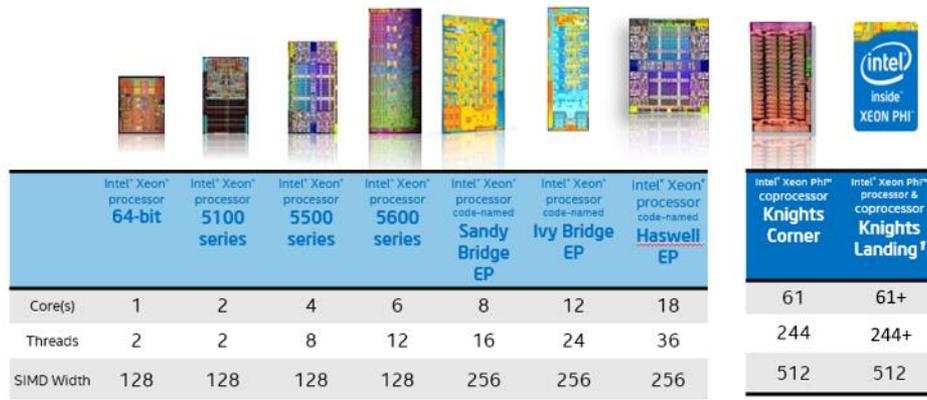
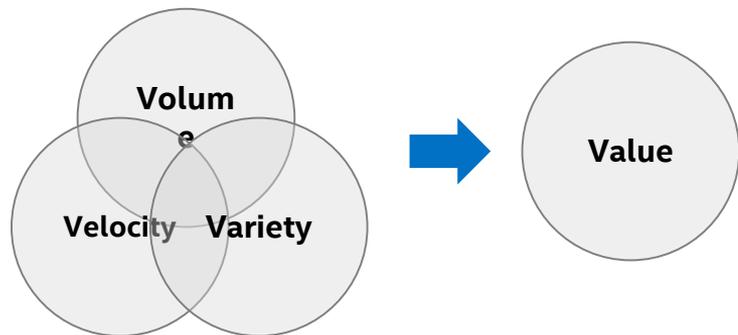
Intel® MKL

Extracting performance from the computing resources

- Core: **vectorization**, prefetching, cache utilization
- Multi-Many core (processor/socket) level **parallelization**
- Multi-socket (node) level **parallelization**
- Clusters **scaling**



Big Data and Machine Learning Challenge



*Product specification for launched and shipped products available on ark.intel.com.
1. Not launched or in planning.

More cores → More Threads → Wider vectors

Problem:

- Big data needs high performance computing.
- Many big data applications leave performance at the table → Not optimized for underlying hardware.

Solution:

- A performance library provides building blocks to be easily integrated into big data analytics workflows.

Intel® Data Analytics Acceleration Library (Intel® DAAL)

An Intel-optimized library that provides building blocks for all data analytics stages, from data preparation to data mining and machine learning

- Python*, Java*, and C++ APIs
- Can be used with many platforms (Hadoop*, Spark*, R*, Matlab*, ...) but not tied to any of them
- Flexible interface to connect to different data sources (CSV, SQL, HDFS, ...)
- Windows*, Linux*, and OS X*
- Developed by same team as the industry-leading Intel® Math Kernel Library
- Open source, Free community-supported and commercial premium-supported options
- Also included in Parallel Studio XE suites



Intel® Threading Building Blocks

Good Tuning Data Gets Good Results

"Using Intel TBB's new flow graph feature, we accomplished what was previously not possible, **parallelize a very sizable task** graph with thousands of interrelationships – all in about a week."

Robert Link
GCAM Project Scientist
Pacific Northwest National Lab

"Intel's TBB was an invaluable help in multi-threading our in-house renderer CGIStudio and is now also used in animation and simulation software. Beside the **ease of use**, it takes care of the two most important aspects of running an application on multiple cores -- **load balancing and scalability.**"

Maurice van Swaaji
Blue Sky Studios

"Intel® TBB provided us with **optimized code** that we did not have to develop or maintain for critical system services. I could assign my developers to code what we bring to the software table."

Michaël Rouillé
CTO
Golaem

[More Case Studies](#)

[Details](#)

Intel® Threading Building Blocks (Intel® TBB)



C++ template library to simplify the task of adding parallelism on a single device or across multiple devices

Specify tasks instead of manipulating threads

- Intel® TBB maps your logical tasks onto threads with full support for nested parallelism

Targets threading for scalable performance

- Uses proven , efficient parallel patterns
- Uses work stealing to support the load balance of unknown execution time for tasks. It has the advantage of low-overhead [polymorphism](#).

Flow graph feature allows developers to easily express dependency and data flow graphs

Has high level parallel algorithms and concurrent containers and low level building blocks like scalable memory allocator , locks and atomic operations.

Commercial support for Intel® Atom™, Core™, Xeon® processors, and for Intel® Xeon Phi™ processors and coprocessors

“Using Intel TBB’s new flow graph feature, we accomplished what was previously not possible, **parallelize a very sizable task graph** with thousands of interrelationships – all in about a week.”

Robert Link
GCAM Project Scientist
Pacific Northwest National Lab

[More Case Studies](#)

Resources and Availability

Intel® Threading Building Blocks (Intel® TBB)

Resources

- Commercial product page: software.intel.com/intel-tbb
- Flow Graph Designer: software.intel.com/articles/flow-graph-designer
- User Forum: software.intel.com/forums/intel-threading-building-blocks

Available on Linux, Windows, macOS and Android

- Commercially available with Intel® Parallel Studio XE 2017: software.intel.com/en-us/intel-parallel-studio-xe
- Community licensing for Intel® Performance Libraries- without Premier support: software.intel.com/nest
- The Open-Source Community Site: www.threadingbuildingblocks.org

Challenges Faced by Developers

- **Performance optimization** is a never-ending task.
- Completing key processing tasks within designated **time constraints** is a critical issue.
- **Hand optimization code** for one platform makes code performance worse on another platform.
- With manual optimization code becomes more **complex and difficult** to maintain.
- Code should run fast as possible **without spending extra effort**.

Different Domains in Intel IPP

Image
Processing

Computer Vision

Color
Conversion

Image Domain

Signal
Processing

Vector Math

Signal Domain

Data
Compression

Cryptography

String
Processing

Data Domain

Intel® Integrated Performance Primitives: Building Blocks for Image, Signal, and Data Processing

Provides developers with ready-to-use functions to accelerate image, signal, data processing and cryptography computation tasks.

Optimized for Intel® Atom™, Core™, and Xeon® processors and for Intel® Xeon Phi™ processors and coprocessors.

License versions available on Linux*, Windows*, macOS*, and Android*

Available as a part of:

- Intel® Parallel Studio XE 2017 - software.intel.com/en-us/intel-parallel-studio-xe
- Community Licensing for Intel® Performance Libraries- without Intel® Premier support: software.intel.com/nest

Correctness Tools Increase ROI by 12%-21%

Cost Factors – Square Project Analysis

CERT: U.S. Computer Emergency Readiness Team, and Carnegie Mellon CyLab

NIST: National Institute of Standards & Technology : Square Project Results

Size and complexity of applications is growing



Correctness tools find defects during development prior to shipment

Reworking defects is 40%-50% of total project effort

Reduce time, effort, and cost to repair

Find errors earlier when they are less expensive to fix

Race Conditions Are Difficult to Diagnose

They Only Occur Occasionally and are Difficult to Reproduce

Correct

Thread 1	Thread 2		Shared Counter
			0
Read count		←	0
Increment			0
Write count		→	1
	Read count	←	1
	Increment		1
	Write count	→	2

Incorrect

Thread 1	Thread 2		Shared Counter
			0
Read count		←	0
	Read count	←	0
Increment			0
	Increment		0
Write count		→	1
	Write count	→	1

Debug Memory and Threading Errors

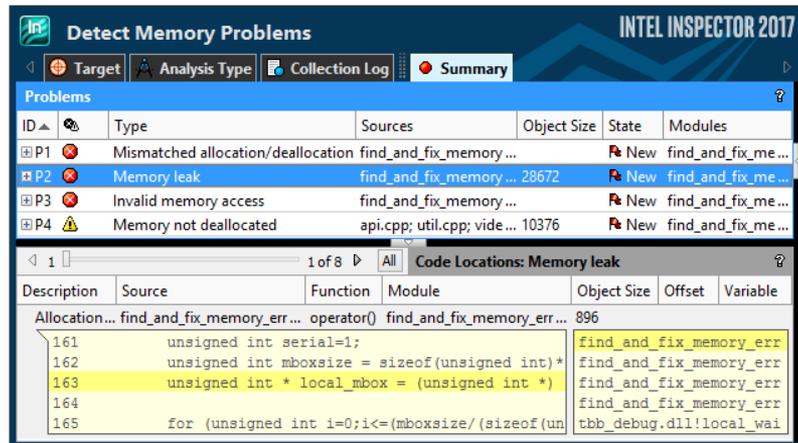
Intel® Inspector

Find and eliminate errors

- Memory leaks, invalid access...
- Races and deadlocks
- C, C++ and Fortran (or a mix)

Simple, Reliable, Accurate

- No special recompiles
- Use any build, any compiler¹
- Analyzes dynamically generated or linked code
- Inspects third-party libraries without source
- Productive user interface + debugger integration
- Command line for automated regression analysis



Clicking an error instantly displays source code snippets and the call stack

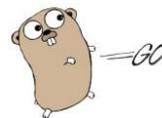
Fits your existing process

¹最適化に関する注意事項 ¹That follows common OS standards.

© 2016 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。

Profile Python* & Go!*



And Mixed Python / C++ / Fortran* – Intel® VTune™ Amplifier

Low-overhead ampling

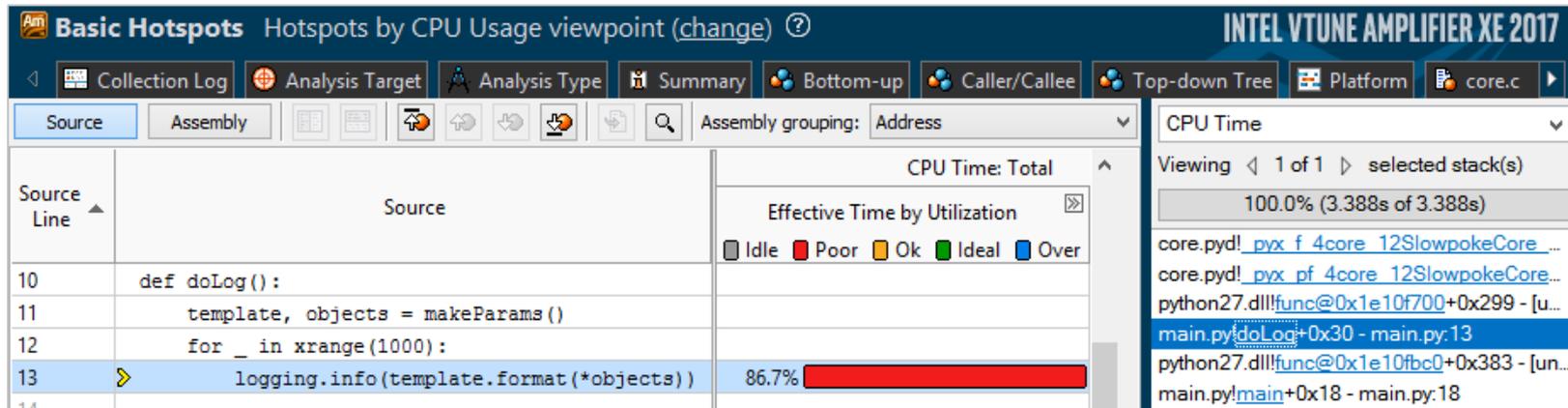
- Accurate performance data without high overhead instrumentation
- Launch application or attach to a running process

Precise line-level details

- No guessing, see source line level detail

Mixed Python/native C, C++, Fortran...

- Optimize native code driven by Python



最適化に関する注意事項

© 2016 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。



Three Keys to HPC Performance

Threading, Memory Access, Vectorization: Intel® VTune™ Amplifier

Threading: CPU utilization

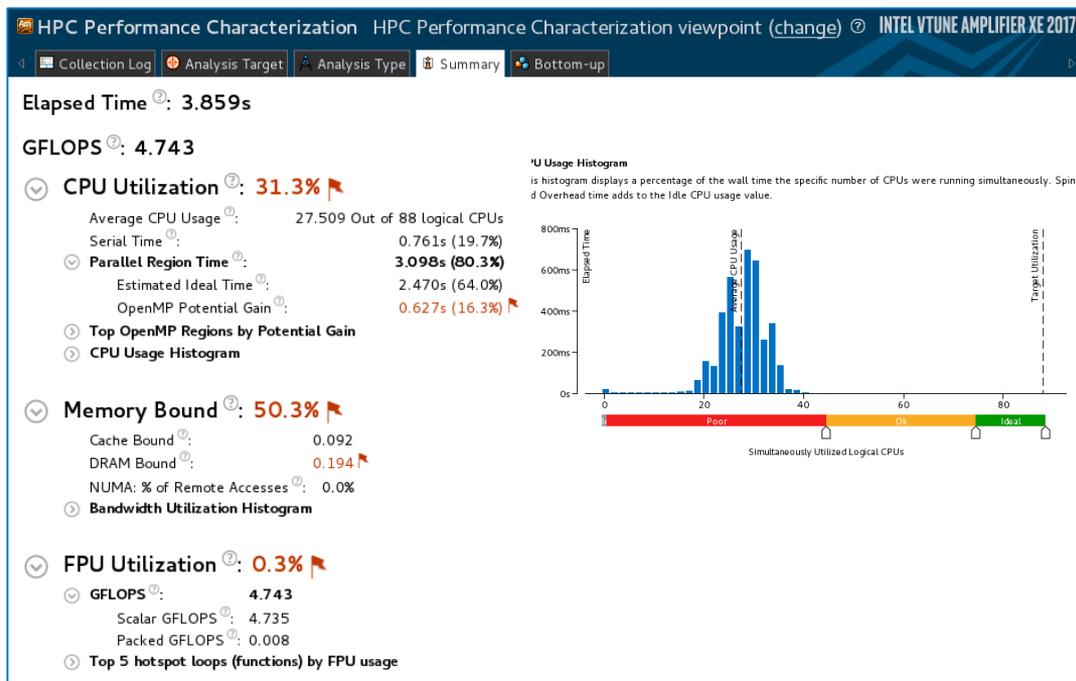
- Serial versus parallel time
- Top OpenMP* regions by potential gain
- Tip: Use hotspot OpenMP region analysis for more detail

Memory access efficiency

- Stalls by memory hierarchy
- Bandwidth utilization
- Tip: Use Memory Access analysis

Vectorization: FPU utilization

- FLOPS[†] estimates from sampling
- Tip: Use Intel® Advisor for precise metrics and vectorization optimization



[†] For 3rd, 5th, 6th Generation Intel® Core™ processors and second generation Intel® Xeon Phi™ processor code named Knights Landing.

Application Performance Snapshot

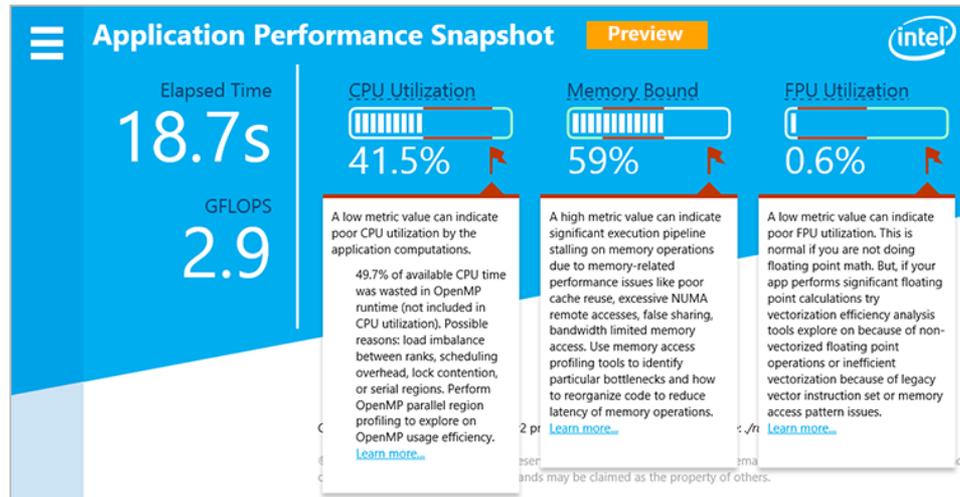
Discover Opportunities for Better Performance with Vectorization and Threading

Objectives

- Simple enough to run during a coffee break
- Highlight where code modernization can help

Users

- Performance teams – fast prioritization of which apps will benefit most
- All Developers – size the potential performance gain from code modernization



Non-Objectives

- Actionable tuning data – that is another tool. Snapshot is just a fast “health” check.

Free download: <http://www.intel.com/performance-snapshot>

Also included with Intel® Parallel Studio and Intel® VTune™ Amplifier products.

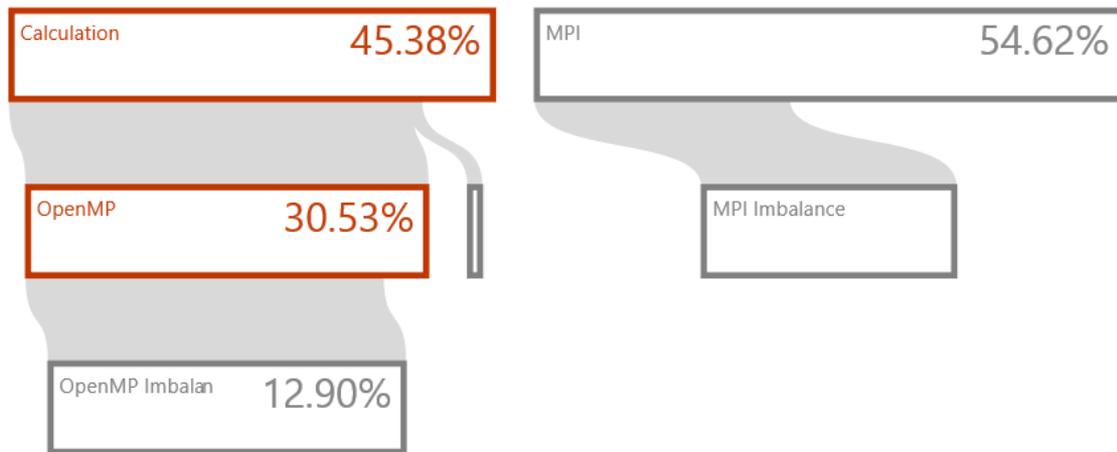
MPI Performance Snapshot

Your application is OpenMP bound.
High OpenMP imbalance has been identified.
Use [Intel VTune Amplifier](#) for further analysis.

Application: /ifs/inn/home/yshchyok/p/svn/testing/ts/results/2015.09.23
12.31.09/itac_testspec/vt_key_default_test_c_icc15_n2_itac_it_mps/test
Number of ranks: 4
Used statistics: app_stat_4p28t.txt, stats_4p28t.txt
Creation date: 2015-09-28 14:58:48

Wallclock time

1.78 sec



TOP 5 MPI functions

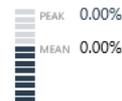
Func	%
Wait	71.98
Barrier	20.92
Init	3.98
Send	2.04
Recv	0.93

GFLOPS

20.67

I/O operations

I/O wait: 0.00 sec 0.00%



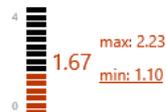
This is the time the application spends waiting for an I/O operation to complete. High percentage of I/O wait time indicates that your application actively reads data from the storage device. This application does not spend much time on I/O operations.

Memory usage



Per-process memory usage affects the application scalability.

Cycles Per Instruction Rate



This could be caused by such issues as memory stalls, instruction starvation, branch misprediction or long latency instructions.

Please use [Intel® VTune™ Amplifier XE](#) to identify the cause of this bottleneck. High values are usually bad. The CPI value may be too high.

Memory Bound Coefficient



It indicates that the application doesn't spend much time waiting for data. High values are usually bad. The application is *not Memory Bound*.

Free download: <http://www.intel.com/performance-snapshot>. Also included with Intel® Parallel Studio Cluster Edition.

最適化に関する注意事項

© 2016 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。

Storage Performance Snapshot

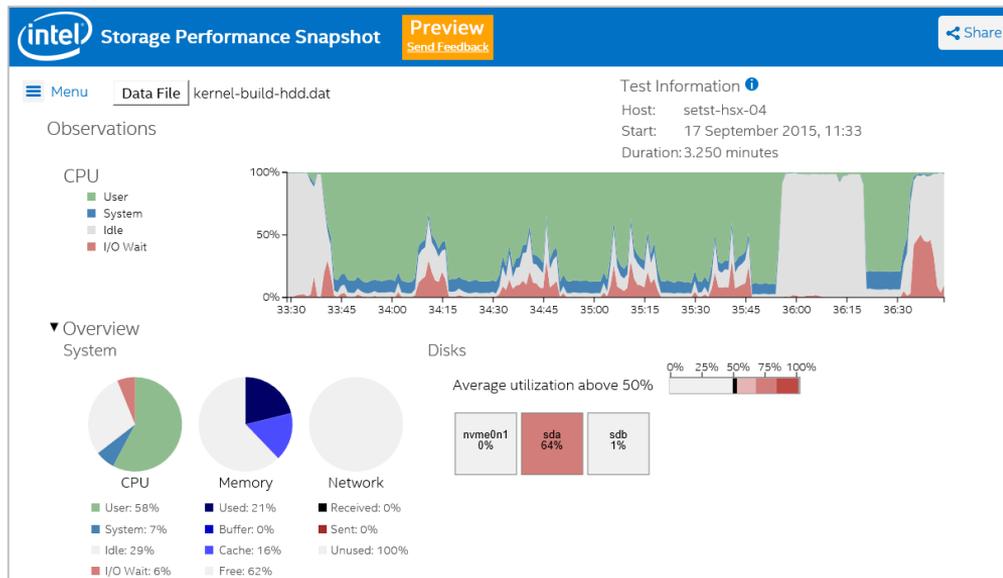
Discover if Faster Storage can Improve Server/Workstation Performance

Learn It On One Coffee Break

- Easy setup
- Quickly see meaningful data
- System view of workload
- Any architecture

Targeted Systems

- Servers and workstations with directly attached storage
- Not scale out storage clusters
- Linux kernel 2.6 or newer
dstat 0.7 or newer
- Windows Server* 2012, Windows* 8, or newer Windows OS



Free download: <http://www.intel.com/performance-snapshot>

Also included with Intel® Parallel Studio and Intel® VTune™ Amplifier products.

Get Faster Code Faster: Intel® Advisor

Vectorization Optimization

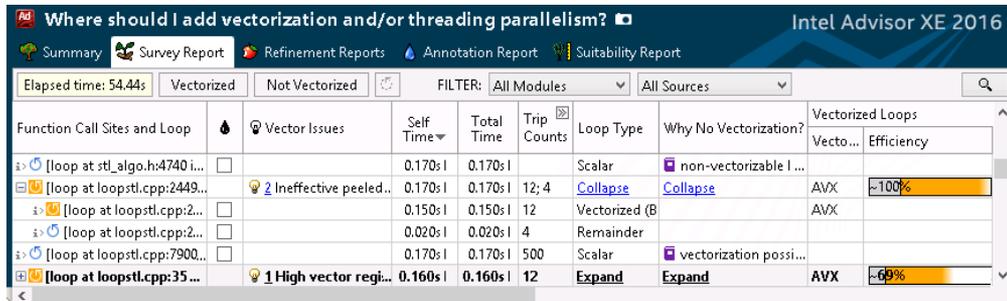
Have you:

- Recompiled for AVX2 with little gain?
- Wondered where to vectorize?
- Recoded intrinsics for new arch.?
- Struggled with compiler reports?

Data-driven vectorization:



- What vectorization will pay off most?
- What's blocking vectorization? Why?
- Are my loops vector friendly?
- Will reorganizing data increase performance?
- Is it safe to just use pragma simd?



Function Call Sites and Loop	Vector Issues	Self Time	Total Time	Trip Counts	Loop Type	Why No Vectorization?	Vectorized Loops
[loop at stl_algo.h:4740 i...		0.170s	0.170s		Scalar	non-vectorizable l ...	
[loop at loopstl.cpp:2449...	2 Ineffective peeled...	0.170s	0.170s	12; 4	Collapse	Collapse	AVX ~100%
[loop at loopstl.cpp:2...		0.150s	0.150s	12	Vectorized (B		AVX
[loop at loopstl.cpp:2...		0.020s	0.020s	4	Remainder		
[loop at loopstl.cpp:7900...		0.170s	0.170s	500	Scalar	vectorization possi...	
[loop at loopstl.cpp:35...	1 High vector regi...	0.160s	0.160s	12	Expand	Expand	AVX ~69%

"Intel® Advisor's Vectorization Advisor permitted me to focus my work where it really mattered. When you have only a limited amount of time to spend on optimization, it is invaluable."

Gilles Civario
Senior Software Architect
Irish Centre for High-End Computing

最適化に関する注意事項

© 2016 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。



Next-Gen Intel® Xeon Phi™ Support

Vectorization Advisor Runs on and Optimizes for Intel® Xeon Phi™

Loops	Vector Issues	Self Time	Loop Type	Vectorized Loops			Instruction Set Analysis			
				Vector ISA	Efficiency	Gain Esti...	VL (V...	Traits	Data Types	
[Loop]	3 Possible i...	35.226s	5.4%	Vectorized+Threaded (Body; Peeled; Re...	AVX512	~28%	2.21x	8	Divisions; FMA; Gathers	Float32; ...
[Loop]	2 Possible in...	26.025s	4.0%	Vectorized (Body)+Threaded (OpenMP)	AVX512			8	Divisions; Gathers; FMA	Float32; ...
[Loop]	1 High vecto...	5.876s		Vectorized (Peeled)+Threaded (OpenMP)	AVX512			8	Divisions; Gathers; FMA	256/512 AVX2; AVX512ER_512; AVX512F...
[Loop]	1 High vecto...	3.324s		Vectorized (Remainder)+Threaded (Open...	AVX512			8	Divisions; Gathers; FMA	256/512 AVX2; AVX512ER_512; AVX512...
[Loop]		34.599s	5.3%	Vectorized (Body; Remainder)	AVX512	~70%	5.64x	8	Divisions; FMA; Square Roots	256/512 AVX2; AVX512ER_512; AVX512...
[Loop]	1 Possible in...	33.849s	5.2%	Vectorized (Body; Peeled; Remainder)	AVX512	~28%	2.24x	8	Divisions; FMA; Gathers	256/512 AVX; AVX2; AVX512ER_512; AV...
[Loop]		19.839s	3.1%	Vectorized (Body; Remainder)	AVX512	72%	11.48x	16; 8		256/512 AVX2; AVX512F_512

AVX-512 ERI – specific to Intel® Xeon Phi

Efficiency (72%), Speed-up (11.5x), Vector Length (16)

Performance optimization problem and advice how to fix it

Issue: Possible inefficient memory access patterns present
Inefficient memory access patterns may result in significant vector code execution slowdown or block automatic vectorization by the compiler. Improve performance by investigating.

Recommendation: Confirm inefficient memory access patterns
There is no confirmation inefficient memory access patterns are present. To confirm: Run a [Memory Access Patterns analysis](#).

Issue: Ineffective peeled/remainder loop(s) present
All or some [source loop](#) iterations are not executing in the [loop body](#). Improve performance by moving source loop iterations from [peeled/remainder](#) loops to the loop body.

Recommendation: Collect trip counts data
The Survey Report lacks [trip counts](#) data that might generate more precise recommendations. To fix: Run a [Trip Counts analysis](#).

Recommendation: Align data

Recommendation: Add data padding
The [trip count](#) is not a multiple of [vector length](#). To fix: Do one of the following:

- Increase the size of objects and add iterations so the trip count is a multiple of vector length.
- Increase the size of static and automatic objects, and use a compiler option to add data padding.

Windows® OS	Linux® OS
/Qopt-assume-safe-padding	-qopt-assume-safe-padding

Program metrics
Elapsed Time: 142.79s
Vector Instruction Set: AVX, AVX2, AVX512, SSE, SSE2
Number of CPU Threads: 4

Loop metrics

Total CPU time	454.08s	100.0%
Time in 88 vectorized loops	41.86s	9.2%

最適化に関する注意事項

Precise, Repeatable FLOPS Metrics

Intel® Advisor: Vectorization Optimization

- FLOPS by loop and function
- All recent Intel® processors (not co-processors)
- Instrumentation (count FLOP) plus sampling (time with low overhead)
- Adjusted for masking with AVX-512 processors

		INTEL ADVISOR 2017						
+ - Function Call Sites and Loops		FLOPS						
		GFLOPS	AI	L1 GB/s	GFLOP	FLOP Per Iteration	L1 GB	L1 Bytes Per Iteration
	[loop in matvec at Multiply.c:69]	0.826 0	0.1633	5.0586	3.0720	32	18.8160	196
	[loop in matvec at Multiply.c:60]	0.912 0	0.1633	5.5853	3.0720	32	18.8160	196
	[loop in matvec at Multiply.c:69]	1.248 0	0.2500	4.9920	1.3440	4	5.3760	16
	[loop in matvec at Multiply.c:60]	1.592 0	0.2500	6.3699	1.3440	4	5.3760	16
	[loop in matvec at Multiply.c:69]	3.055	0.2500	12.2205	0.0960	16	0.3840	64
	[loop in matvec at Multiply.c:60]	6.282	0.2500	25.1279	0.0960	16	0.3840	64

最適化に関する注意事項

© 2016 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。



Enhanced Memory Access Analysis: Intel® Advisor

Are you Bandwidth or Compute Limited?

Measure footprint

- Compare to cache size
- Does it fit in cache?

Variable references

- Map data to variable names for easier analysis

Gather/scatter

- Detect unneeded gather/scatters that reduce performance

Site Location	Loop-Carried Dependencies	Strides Distribution ▲	Access Pattern	Max. Site Footprint ▲
[loop in s4117_ at loopstl.cpp:76...	No information available	50% / 50% / 0%	Mixed strides	192B
[loop in s442_ at loopstl.cpp:6815]	No information available	56% / 0% / 44%	Mixed strides	256B
[loop in s272_ at loopstl.cpp:3447]	No information available	60% / 0% / 40%	Mixed strides	320B

ID	Stride	Type	Source	Nested Function	Variable references	Access Footprint	Mo
P2		Gather stride	loopstl.cpp:3450		a, c, d	320B	lcd_

```

3448     if (e[i_] >= *t)
3449     {
3450         a[i_] += c[i_] * d[i_];
3451         b[i_] += c[i_] * c[i_];
3452     }

```

File: cache_8ca80efbe6ecb40b7d2e3f3cf0d5d6ff_loopstl.cpp:3450

Line	Source	Stride
3450	a[i_] += c[i_] * d[i_];	[1]
3451	b[i_] += c[i_] * c[i_];	
3452	}	
3453	dummy_(1d, n, sa[1], sb[1], sc_[1], sd_	

Address	Line	Assembly	Physical Stride
0x43265a	3450	vgatherdpsz (%r8,%zmm0,4), %k1, %zmm2	[1]
0x432661	3403	leaq (%r13,%rsi,1), %r8	
0x432666	3450	vgatherdpsz (%r9,%zmm8,4), %k3, %zmm1	[1]

Details View

Gather (irregular) access

Operand Size (bits): 32
 Operand Type: bit*16,float32*16
 Vector Length: 16
 Memory access footprint: 320B

Gather/scatter details

Pattern: "Unit"
 Instruction accesses values in contiguous memory throughout the loop:
 - unit stride within instruction
 - stride between iterations = vector length

Horizontal stride (bytes): 4
 Vertical stride (bytes): 64

最適化に関する注意事項

© 2016 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。



Start Tuning for AVX-512* without AVX-512 hardware

Intel® Advisor: Vectorization Advisor

Use `-axCOMMON-AVX512 -xAVX` compiler flags to generate both code-paths

- AVX(2) code path (executed on Haswell and earlier processors)
- AVX-512 code path for newer hardware

Compare AVX and AVX-512 code with Intel Advisor

Loops	🔥	Self Time	Loop Type	Vectorized Loops				Instruction Set Analysis				Advanced	
				Vect...	Efficiency	Gain...	VL (...)	Compiler Es...	Traits	Data T...	Vector W...	Instruction Sets	Vectorization D...
[-] [loop in s352_at loopstl.cpp:5939]	<input type="checkbox"/>	0,641s	Vectorized (Body)	AVX2	~54%	2,15x	4	2,15x	FMA; Inserts	Float32	128	AVX; FMA	
[-] [loop in s352_at loopstl.cpp:5939]	<input type="checkbox"/>	n/a	Remainder [Not Executed]				4		FMA				
[+] [loop in s352_at loopstl.cpp:5939]	<input type="checkbox"/>	0,641s	Vectorized (Body)	AVX2			4	2,15x	Inserts; FMA				
[-] [loop in s352_at loopstl.cpp:5939]	<input type="checkbox"/>	n/a	Vectorized (Body) [Not Executed]	AVX512			16	3,20x	Gathers; FMA				
[-] [loop in s352_at loopstl.cpp:5939]	<input type="checkbox"/>	n/a	Vectorized (Remainder) [Not Executed]	AVX512			16	2,70x	Gathers; FMA				
[-] [loop in s125_ ASomp\$parallel_for@...]	<input type="checkbox"/>	0,496s	Vectorized Versions	AVX2	~100%	13,54x	8	<13,54x	FMA; NT-stores				
[-] [loop in s125_ ASomp\$parallel_for...]	<input type="checkbox"/>	n/a	Peeled [Not Executed]				8		FMA				
[-] [loop in s125_ ASomp\$parallel_for...]	<input type="checkbox"/>	n/a	Remainder [Not Executed]				8		FMA				
[+] [loop in s125_ ASomp\$parallel_for...]	<input type="checkbox"/>	0,465s	Vectorized (Body)	AVX2			8	13,54x					
[-] [loop in s125_ ZSomp\$parallel_for...]	<input type="checkbox"/>	n/a	Vectorized (Peeled) [Not Executed]	AVX512			16	6,77x	FMA				
[-] [loop in s125_ ZSomp\$parallel_for...]	<input type="checkbox"/>	n/a	Vectorized (Body) [Not Executed]	AVX512			32	30,61x	NT-stores				
[-] [loop in s125_ ZSomp\$parallel_for...]	<input type="checkbox"/>	n/a	Vectorized (Remainder) [Not Executed]	AVX512			16	9,78x	FMA				

Inserts (AVX2) vs.
Gathers (AVX-512)

Speed-up estimate:
13.5x (AVX2) vs.
30.6x (AVX-512)

Faster Code Faster Using Intel® Advisor

Vectorization

"Intel® Advisor's Vectorization Advisor permitted me to focus my work where it really mattered. When you have only a limited amount of time to spend on optimization, it is invaluable."

Gilles Civario
Senior Software Architect
Irish Centre for High-End Computing

"Intel® Advisor's Vectorization Advisor fills a gap in code performance analysis. It can guide the informed user to better exploit the vector capabilities of modern processors and coprocessors."

Dr. Luigi Iapichino
Scientific Computing Expert
Leibniz Supercomputing Centre

Threading

"Intel® Advisor has been extremely helpful in identifying the best pieces of code for parallelization. We can save several days of manual work by targeting the right loops and we can use Advisor to find potential thread safety issues to help avoid problems later on."

Carlos Boneti
HPC software engineer,
Schlumberger

"Intel® Advisor has allowed us to quickly prototype ideas for parallelism, saving developer time and effort, and has already been used to highlight subtle parallel correctness issues in complex multi-file, multi-function algorithms."

Simon Hammond
Senior Technical Staff
Sandia National Laboratories

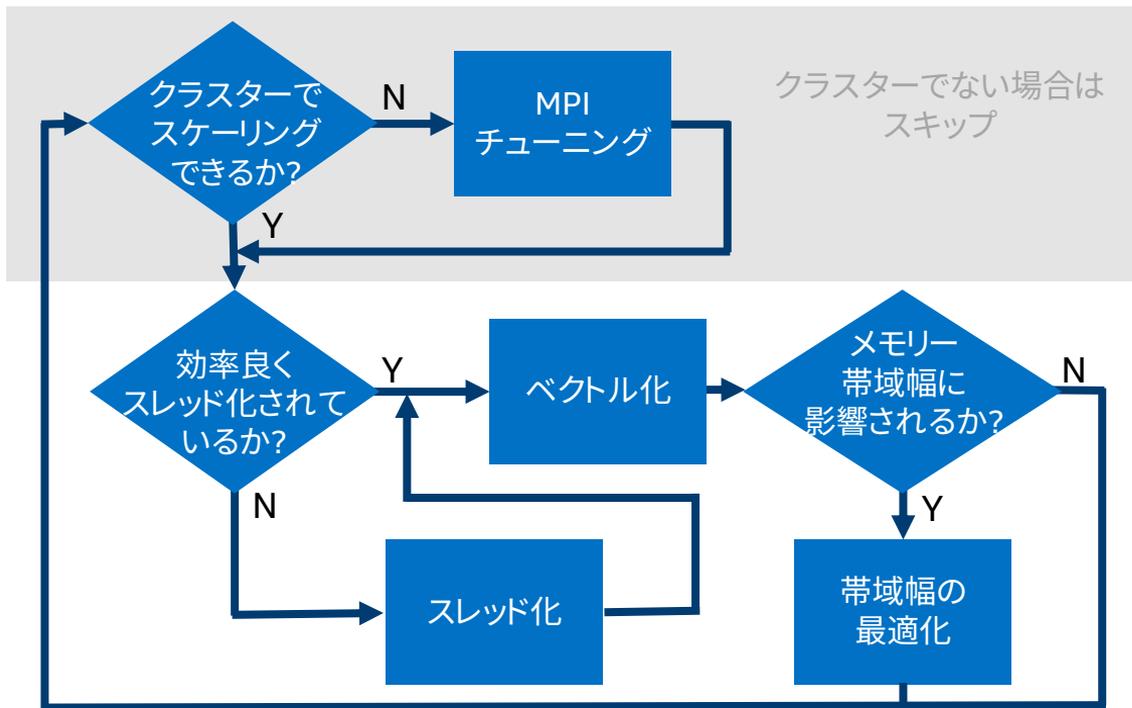
[More Case Studies](#)



どのツールを使用すべきか？

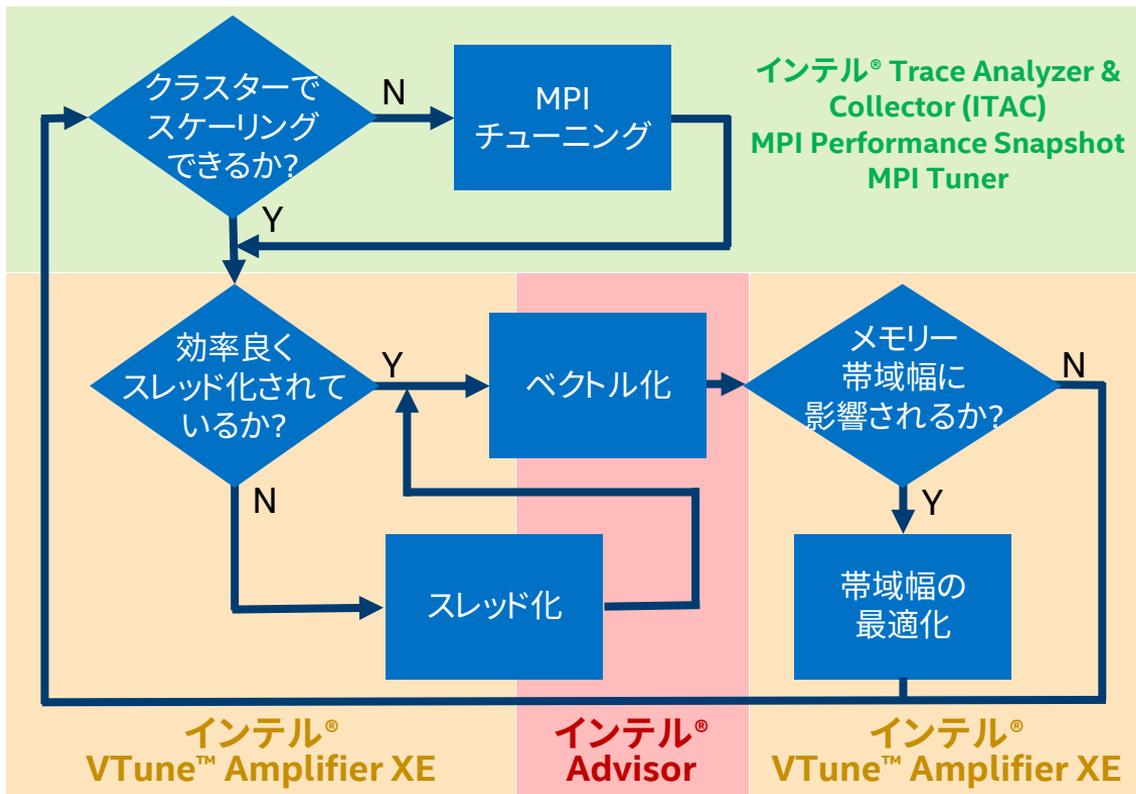
並列ハードウェア上でのパフォーマンスの最適化

繰り返し作業...



診断を支援するパフォーマンス解析ツール

インテル® Parallel Studio XE



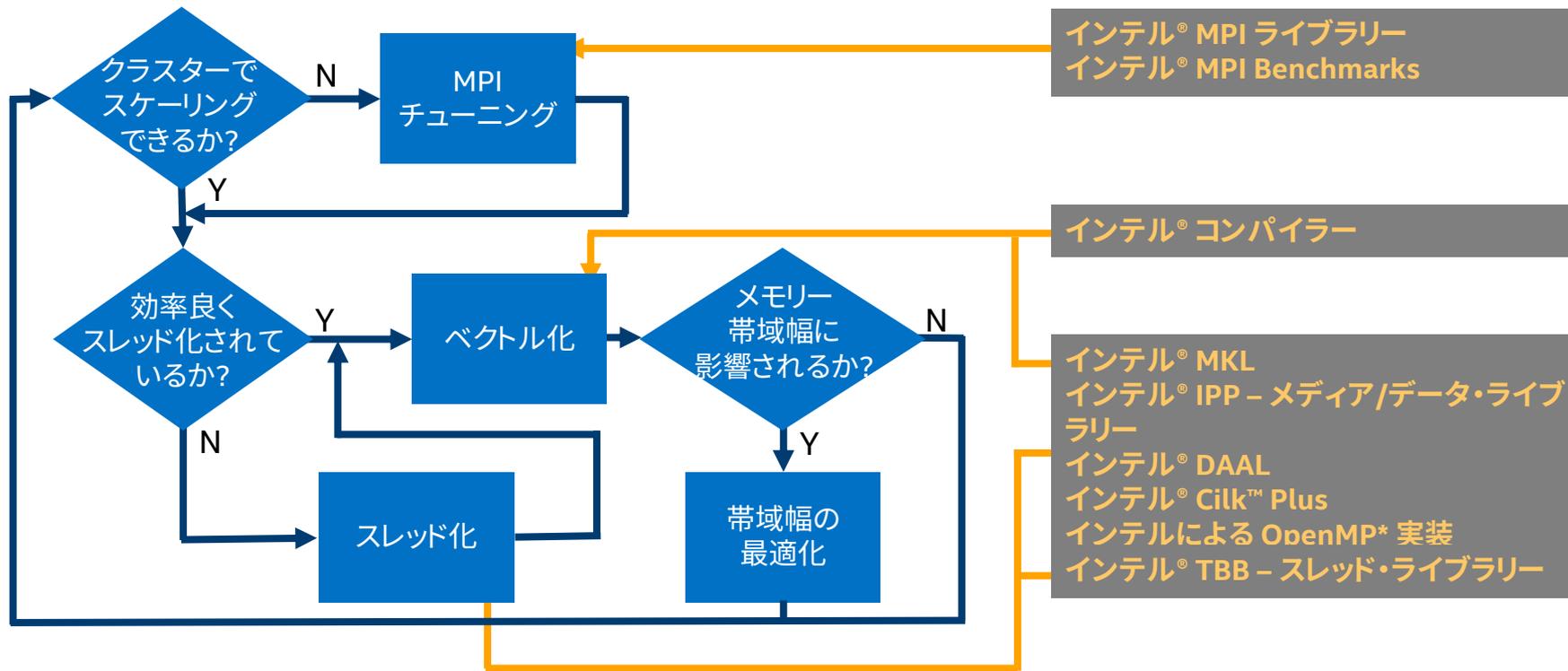
最適化に関する注意事項

© 2016 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。

ハイパフォーマンスな実装を支援するツール

インテル® Parallel Studio XE



最適化に関する注意事項

© 2016 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。

問題サイズとシステム構成情報

インテル® Distribution for Python* ベンチマーク

Hardware/Problem Size	dot	lu	det	inv	cholesky	fft
Intel® Xeon® processor (32 core) and Intel® Xeon Phi™ processor (64 core)	(20k, 10k) and (10k, 20k)	(35k, 35k)	(15k, 15k)	(25k, 25k)	(40k, 40k)	520k
Intel Xeon processor (1 core)	(20k, 5k) and (5,20k)	(20k, 20k)		(10k, 10k)		
Intel Xeon Phi processor (1 core)	(20k, 300) and (300, 20k)	(6k, 6k)	(4k, 4k)	(2k, 2k)	(10k, 10k)	

Configuration Info: apt/atlas: installed with apt-get, Ubuntu* 16.10, Python* 3.5.2, numpy* 1.11.0, scipy* 0.17.0; pip*/openblas*: installed with pip, Ubuntu 16.10, python 3.5.2, numpy 1.11.1, scipy 0.18.0; Intel Python: Intel® Distribution for Python 2017;. Hardware: Intel Xeon processor: Intel Xeon processor E5-2698 v3 @ 2.30 GHz (2 sockets, 16 cores each, HT=off), 64 GB of RAM, 8 DIMMS of 8GB@2133MHz; Intel Xeon Phi processor: Intel Intel® Xeon Phi™ processor 7210 1.30 GHz, 96 GB of RAM, 6 DIMMS of 16GB@1200MHz

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. * Other brands and names are the property of their respective owners. Benchmark source: Intel Corporation.

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804

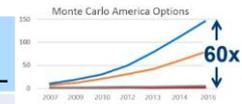
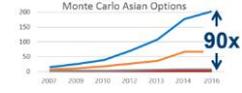
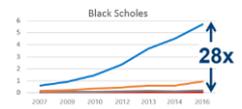
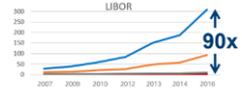
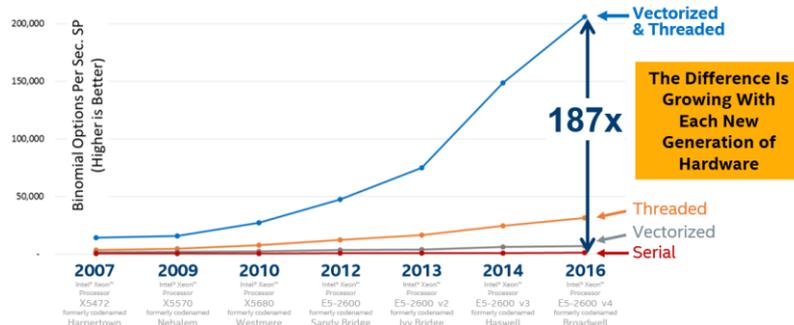
最適化に関する注意事項

© 2016 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。



2007年～2016年のベンチマークのシステム構成



プラットフォーム、ハードウェア、ソフトウェア

プラットフォーム	スケーリングされていないコアクロックの周波数	コア/ソケット	ソケット数	L1 データ キャッシュ	L2 キャッシュ	L3 キャッシュ	メモリー	メモリー周波数	メモリーアクセス	H/Wプリフェッチ有効	HT有効	ターボ有効	Cステート	OS	カーネル	コンパイラー
Intel® Xeon® プロセッサ 5472	3.00GHz	4	2	32K	6MB	なし	32GB	800MHz	UMA	Y	N	N	無効	Fedora* 20	3.11.10-301.fc20	icc 14.0.1
Intel® Xeon® プロセッサ X5570	2.90GHz	4	2	32K	256K	8MB	48GB	1333MHz	NUMA	Y	Y	Y	無効	Fedora* 20	3.11.10-301.fc20	icc 14.0.1
Intel® Xeon® プロセッサ X5680	3.33GHz	6	2	32K	256K	12MB	48 MB	1333 MHz	NUMA	Y	Y	Y	無効	Fedora* 20	3.11.10-301.fc20	icc 14.0.1
Intel® Xeon® プロセッサ E5-2690	2.90GHz	8	2	32K	256K	20MB	64 GB	1600MHz	NUMA	Y	Y	Y	無効	Fedora* 20	3.11.10-301.fc20	icc 14.0.1
Intel® Xeon® プロセッサ E5-2697 v2	2.70GHz	12	2	32K	256K	30MB	64 GB	1867MHz	NUMA	Y	Y	Y	無効	RHEL 7.1	3.10.0-229.el7.x86_64	icc 14.0.1
Intel® Xeon® プロセッサ E5-2600 v3	2.20GHz	18	2	32K	256K	46MB	128 GB	2133 MHz	NUMA	Y	Y	Y	無効	Fedora* 20	3.13.5-202.fc20	icc 14.0.1
Intel® Xeon® プロセッサ E5-2600 v4	2.30GHz	18	2	32K	256K	46MB	256GB	2400MHz	NUMA	Y	Y	Y	無効	RHEL 7.0	3.10.0-123.el7.x86_64	icc 14.0.1
Intel® Xeon® プロセッサ E5-2600 v4	2.20GHz	22	2	32K	256K	56MB	128GB	2133MHz	NUMA	Y	Y	Y	無効	CentOS* 7.2	3.10.0-327.el7.x86_64	icc 14.0.1

Key:

- Vectorized & Threaded
- Threaded
- Vectorized
- Serial

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依存の最適化は、インテル® マイクロプロセッサ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。注意事項の改訂 #20110804

インテル社内での測定値。

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一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。

© 2016 Intel Corporation. 無断での引用、転載を禁じます。Intel、インテル、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Inte Atom、Intel Core、Xeon、Intel Xeon Phi、Cilk、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 の商標です。Microsoft、Visual Studio、Windows、および Windows Server は、米国 Microsoft Corporation の、米国およびその他の国における登録商標または商標です。OpenCL および OpenCL ロゴは、Apple Inc. の商標であり、Khronos の使用許諾を受けて使用しています。

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一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一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一に関して、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一依存の最適化は、インテル® マイクロプロセッサ一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一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。

注意事項の改訂 #20110804

