インテル® System Studio 2019 の最新情報

久保寺　陽子

エンドツーエンドのデータ中心のストラテジー: デバイスからクラウドまで

垂直ソリューションの作成 + 水平プラットフォームと製品の開発 + 優れたエコシステムの構築

モノ

ネットワーク

クラウド

インテル® ソフトウェアはすべてをつなぐ

インテル® System Studio

スマート･コネクテッド･デバイス向けシステム/IoT アプリケーションを容易に開発

インテル® System Studio とは

システム/IoT アプリケーションの開発をスピードアップすることを目的として開発された、使いやすく包括的なクロスプラットフォームのツールスイート

最適化コンパイラー、高度にチューニングされたライブラリー、解析ツール、デバッグツール、高度なクラウドコネクター、400 を超えるセンサーへのアクセスを提供

対象ユーザー

- **すべての業種の組込み/IoT アプリケーション開発者**
- **デバイスメーカー**
- **システム･インテグレーター**

使用すべき理由

競争優位性を得るため

- 製品の開発期間を短縮
- パフォーマンスと電力効率を向上
- システムの信頼性を強化

年間 B2B IoT 収益予想は 3000 億ドル以上 – 産業部門をリード[1]

ダウンロード: isus.jp/intel-system-studio/

[1]Forbes、2017 年 12 月 10 日 (英語)

さらなるパフォーマンスによってもたらされるビジネス上の利点

- インテル® ハードウェアの性能を最大限に引き出して**さらなるパフォーマンス**を達成
- **効率**と**開発者の生産性**を向上
- **ビジネスの課題を解決し、イノベーションを促進**
- **さまざまなプラットフォームとデバイスでスケーリング**、エンドツーエンドの互換性と相互運用性を向上

インテル® System Studio の利点

- システムとアプリケーションのパフォーマンスを向上
- 簡単なシステム起動、容易なシステム検証、トレースの自動化
- 電力効率の解析と改善

インテル® System Studio 2019 の新機能

1 つの包括的なツールスイートでシステム/IoT 開発をイノベーション

パフォーマンスと電力効率を向上

- **新しいインテル® Advisor** は、ハードウェアのパフォーマンスを最大限に引き出せるように、アプリケーションのベクトル化とスレッド化の向上を支援します。
- **新しい OpenCL* ツール**は、インテル® プラットフォーム (CPU/GPU) 上での OpenCL* アプリケーションの開発を支援します。OpenCL* カーネル対応の高度なコンパイラー、ランタイムデバッガー、パフォーマンス・アナライザーを使用して、計算負荷が高いワークロードをインテル® プロセッサー・グラフィックスにオフロードします。
- **コードのベクトル化とスレッド化を解析する新しい解析ツール。**インテル® Advisor には、アプリケーションがハードウェアのパフォーマンスを最大限に引き出せるように支援するベクトル化解析ツールとスレッド化設計ツールが含まれています。
- **プラットフォームレベルのパフォーマンスの詳細な情報を提供。**インテル® VTune™ Amplifier を使用して、プラットフォームレベルのメモリー/ストレージのボトルネックとインバランスを識別できます。新しい単純化されたワークフローにより、さらに使いやすくなりました。
- **ハードウェア・サポート:** 第 8 世代インテル® Core™ U シリーズ・プロセッサー・ファミリーおよびインテル® Core™ S シリーズ・プロセッサー・ファミリー、インテル® Xeon® プロセッサー D-1500 製品ファミリー

スマート・コネクテッド・デバイスの開発を高速化

- **Wind River* Linux*/Yocto Project* 向けの新しい Eclipse* プラグイン** - より簡単なシステム開発とクロスビルド SDK インポート機能を考慮した、Yocto Project* 互換ターゲットのアプリケーションおよびプラットフォーム・プロジェクトを作成/インポートします。
- **65 の新しいコードウィザードとサンプルコード**により開発時間を短縮します。

システムの信頼性を強化

- **インテル® Debug Extensions for WinDbg** は、インテル® Processor Trace と Direct Connect Interface (DCI) のタイムスタンプにより、タイミング問題とパフォーマンス問題の優先順位付けを支援します。ターゲットと接続タイプを簡単にセットアップした後、複数の同時デバッグ/トレース接続を管理してシステム開発を迅速化します。コマンドラインで実行するシステムトレースを設定および自動化します。

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。

インテル® System Studio 2019 のコンポーネント

Ultimate Edition[1]

デバッグ
- アプリケーションとシステム
- デバッグ/トレース

解析
- 電力とパフォーマンス
- ベクトル化、メモリーとスレッド

ビルド
- インテル® C/C++ コンパイラー
- OpenCL* ツール
- ライブラリー 画像、信号、数学、スレッド、データ解析 IoT 接続ツール

UEFI*[3] エージェント

JTAG、JTAG over USB

Linux*[2]、Android*、Windows*、VxWorks* 上で動作するシステムおよびアプリケーション・コード

統合開発環境 (IDE): Eclipse* Standard、Microsoft* Visual Studio*、Wind River* Workbench
ホスト OS サポート: Linux*、Windows*、macOS*

インテル® アーキテクチャー・ベースのプラットフォーム

[1] インテル® System Studio には 3 つのエディションがあります。有償ライセンスはプライオリティー・サポートが提供されます。無償ライセンスはコミュニティーによるサポートが提供されます。
[2] Linux*、組込み Linux*、Wind River* Linux*、Yocto Project*
[3] UEFI*: Unified Extensible Firmware Interface

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。

インテル® System Studio 2019 と インテル® Parallel Studio 2019 の違い

	インテル® System Studio									インテル® Parallel Studio XE						
	Composer Edition			Professional Edition			Ultimate Edition			Composer Edition			Professional Edition			Cluster Edition
インテル® C++ コンパイラー	■	■	■	■	■	■	■	■	■	■	■		■	■		■
インテル® (Visual) Fortran コンパイラー										■		■	■		■	■
インテル® Distribution for Python*										■	■	■	■	■	■	■
インテル® MKL	■		■	■		■	■		■	■	■	■	■	■	■	■
インテル® DAAL	■	■	■	■	■	■	■	■	■	■	■		■	■		■
インテル® IPP	■	■	■	■	■	■	■	■	■	■	■		■	■		■
インテル® TBB	■	■	■	■	■	■	■	■	■	■	■		■	■		■
インテル® Cilk™ Plus										■	■		■	■		■
OpenMP*										■	■	■	■	■	■	■
インテル拡張 GDB アプリケーション・デバッガー	■	■		■	■		■	■								
インテル® Debugger for Heterogeneous Compute	■			■			■									
インテル® VTune™ Amplifier ※4				■	■	■	■	■	■				■	■	■	■
インテル® Energy Profier				■	■	■	■	■	■							
インテル® Graphics Performance Analyzer				■	■	■	■	■	■							
インテル® Inspector				■		■	■		■				■	■	■	■
インテル® System Debugger (JTAG) ※2							■	■	■							
インテル® Debugger Extensions for WinDbg									■							
インテル® Advisor				■	■	■	■	■	■				■	■	■	■
インテル® MPI ライブラリー																■
インテル® Trace Analyzer & Collector																■
インテル® Cluster Checker																■

インテル® System Studio コンポーネント・ツールと機能

デバッグ

インテル® System Debugger
インテル® Debug Extensions for WinDbg

解析

インテル® VTune™ Amplifier
電力解析
インテル® Advisor
インテル® Inspector
インテル® Graphics Performance Analyzers

ビルド

インテル® C++ コンパイラー
インテル® MKL
インテル® DAAL
インテル® IPP
インテル® TBB
OpenCL* ツール
IoT 接続ツール

インテル® System Debugger 2019 の新機能

システムの信頼性を向上、開発期間を短縮

インテル® プラットフォームの詳細なシステムワイドのデバッグとトレースにより、システムの信頼性を向上し、開発を高速化

新しいプロセッサーのサポート

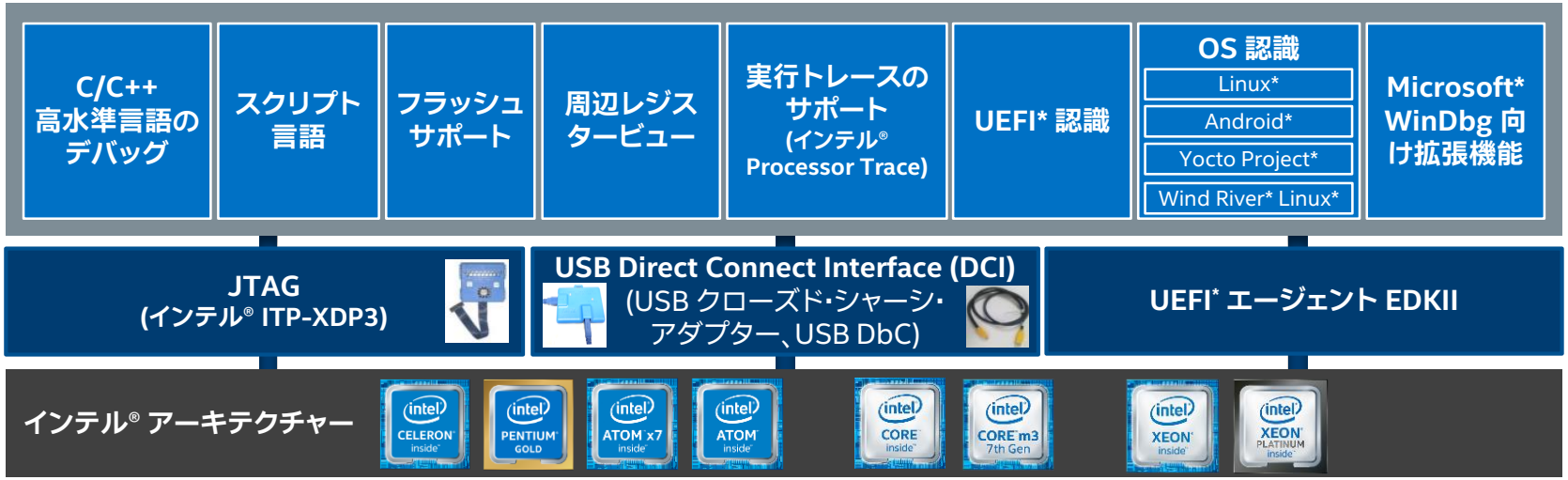
- 第 8 世代インテル® Core™ U シリーズ・プロセッサー・ファミリーおよびインテル® Core™ S シリーズ・プロセッサー・ファミリー
- インテル® Xeon® プロセッサー D-1500 製品ファミリー

システムの信頼性を強化

- インテル® Processor Trace と Direct Connect Interface (DCI) のタイムスタンプにより、タイミング問題とパフォーマンス問題の優先順位付けを支援する、インテルが提供する Microsoft* の WinDbg カーネルデバッガー向け拡張機能が利用可能
- ターゲット接続アシスタントによりインテル® プラットフォーム・ベースのターゲットと接続タイプを簡単にセットアップして複数の同時デバッグ/トレース接続を管理
- 新しい Python* スクリプト・インターフェイスによりコマンドラインで実行するシステムトレースを設定および自動化
- 単純化されたカスタムトレース設定のセットアップによりプラットフォーム・トレースを素早く開始 (プレビュー)

インテル® System Debugger

インテル® アーキテクチャー・ベースのシステム・ソフトウェア/IoT アプリケーションのデバッグとトレース

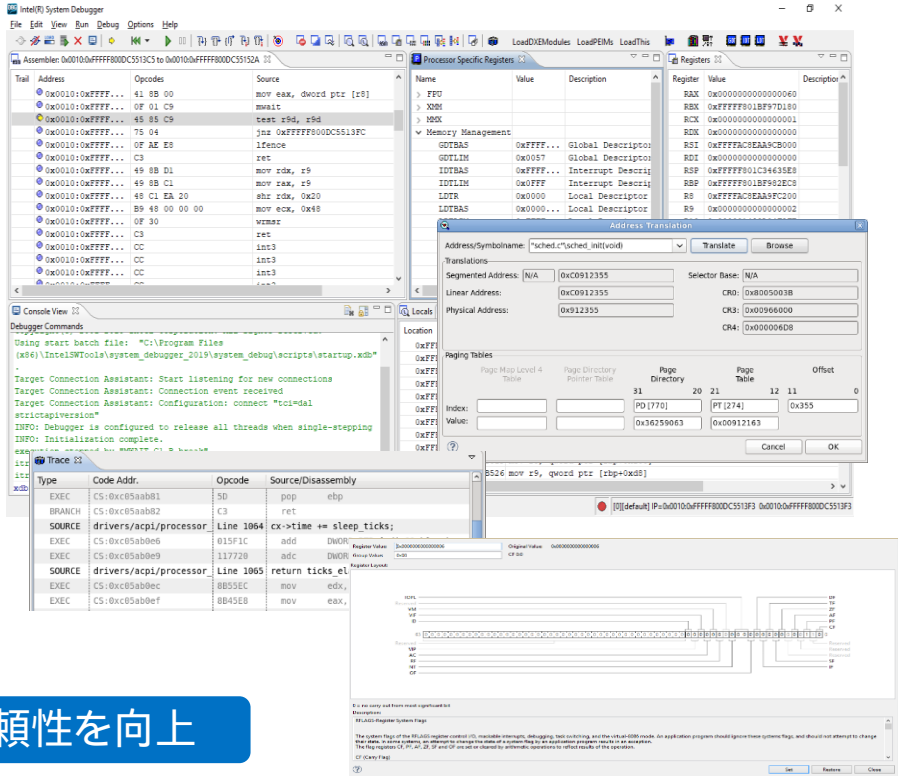


- ハードウェアとソフトウェアの詳細な情報によりデバッグと検証をスピードアップして、開発期間を短縮
- ファームウェア、UEFI/BIOS、オペレーティング・システム、デバイスドライバーのプラットフォーム起動とデバッグに最適
- Windows* と Linux* のカーネルソース、動的にロードされるドライバーとカーネルモジュールのデバッグ

インテル® System Debugger – システムデバッグ

総合的なシステムワイドのハードウェア/ソフトウェアのデバッグ

- フル機能を備えたソースレベルのデバッガーと**インテル® ハードウェア・アーキテクチャーの詳細な情報**により複雑なシステムの問題を素早く解決できるように支援
- CPU レジスター、モデル固有レジスター、その他のハードウェア・ステートに**直接アクセスして変更**
- 割り込み/グローバル・ディスクリプター・テーブルなどの**低レベルのデータ構造に簡単にアクセス**
- **コマンドライン・スクリプト・インターフェイス**を使用してマシンステートとプログラムステートの両方を直接操作するスクリプトを実行
- **低コストのダイレクト USB 接続**または従来の JTAG プローブを使用してデバッグ

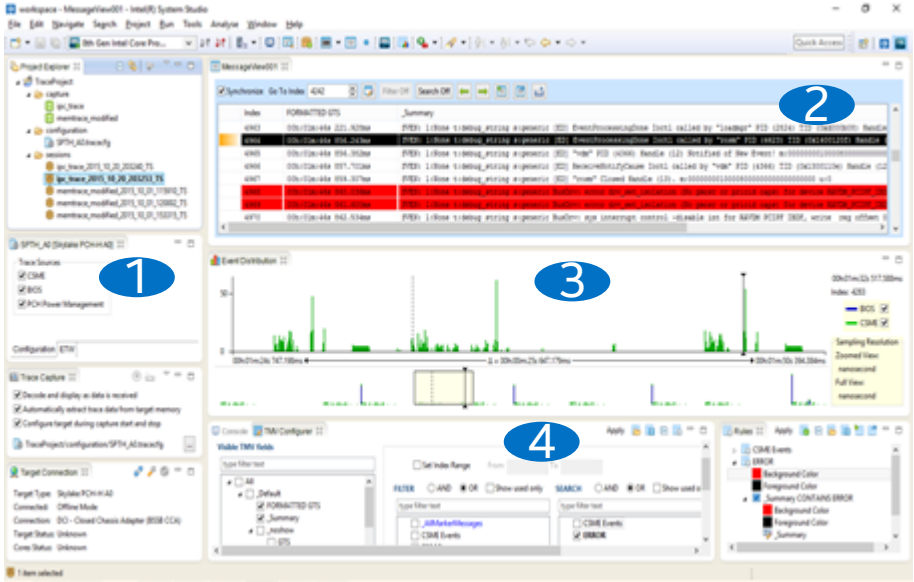


システムの起動時間を高速化し、システムの信頼性を向上

インテル® System Debugger – システムトレース

総合的なシステムワイドのハードウェア/ソフトウェア・イベント・トレース

- ハードウェアとソフトウェア間の**複雑な相互作用を素早く解析**して、タイムスタンプ付きの関連付けされたトレース情報により、複雑なシステムの問題を効率良く特定して解決
- ソフトウェア、ファームウェア、ハードウェア・コンポーネントからの**システムワイドのトレースを集約して視覚化** (例: BIOS、ME、アーキテクチャー・イベント・トレース、Microsoft* ETW イベント)
- USB3 ポート経由のインテル® DCI 接続を利用して**トレースデータに簡単にアクセス**
- **1 つの普遍的なトレースログ**を使用して異なるコンポーネントにわたるシステムの問題解決に協力して取り組める



1. トレースソースの設定
2. 詳細なトレース・メッセージ・ビュー
3. クイック・ナビゲーションを可能にするイベント分布ビュー
4. さまざまなフィルターと検索機能

システムトレースにより複雑なシステムの問題を素早く特定、切り分け、解決

インテル® System Studio コンポーネント・ツールと機能

デバッグ

インテル® System Debugger
インテル® Debug Extensions for WinDbg

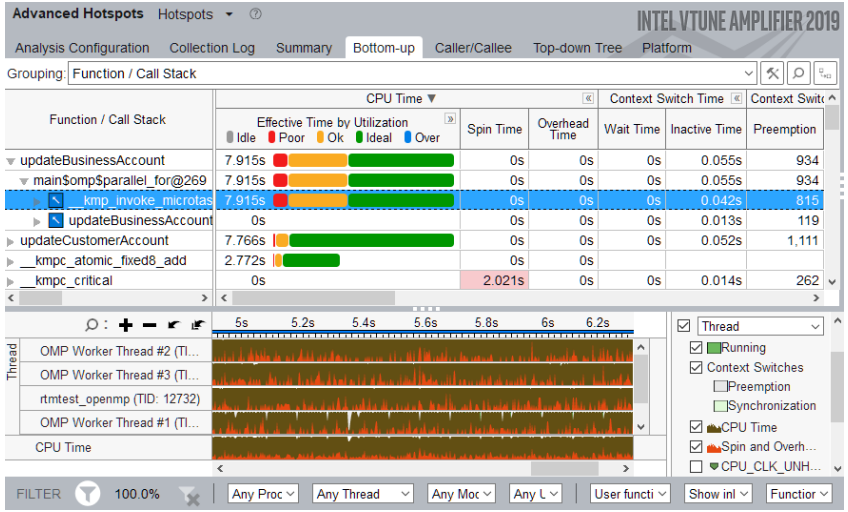
解析

インテル® VTune™ Amplifier
電力解析
インテル® Advisor
インテル® Inspector
インテル® Graphics Performance Analyzers

ビルド

インテル® C++ コンパイラー
インテル® MKL
インテル® DAAL
インテル® IPP
インテル® TBB
OpenCL* ツール
IoT 接続ツール

インテル® VTune™ Amplifier パフォーマンス・プロファイラーによる アプリケーションのパフォーマンスとスケーラビリティーの解析とチューニング



高速でスケーラブルなコードを迅速に開発

- C、C++、Java*、Python*、Go* に加えて、これらの言語が混在したコードを正確にプロファイル
- CPU/GPU、スレッド、メモリー、キャッシュ、ストレージほかを最適化
- 時間を節約: 詳細な解析により洞察が得られる

バージョン 2019 の新機能 (ハイライト)

- チューニングをより簡単に行えるようにワークフローを単純化
- I/O 解析 - SPDK ストレージと DPDK ネットワーク・パフォーマンスのチューニング
- 新しいプラットフォーム・プロファイラー - プラットフォームレベルのパフォーマンスの詳細な情報により、メモリー/ストレージのボトルネックとインバランスを識別

インテル® VTune™ Amplifier - バージョン 2019 の新機能の詳細

パフォーマンス・プロファイラー

さらに使いやすく

- 分かりやすいチューニング・ワークフローと単純化されたセットアップ
- 新しい視覚化により全般的な複雑なカウンター情報を単純化

I/O 解析 - SPDK ストレージと DPDK ネットワーク・パフォーマンスのチューニング

- 実際のワークが行われていない “空の” ポーリングサイクルを測定

プラットフォーム・プロファイラー

- 長いデータ収集によりハードウェア構成の問題点と適切にチューニングされていないアプリケーションを検出

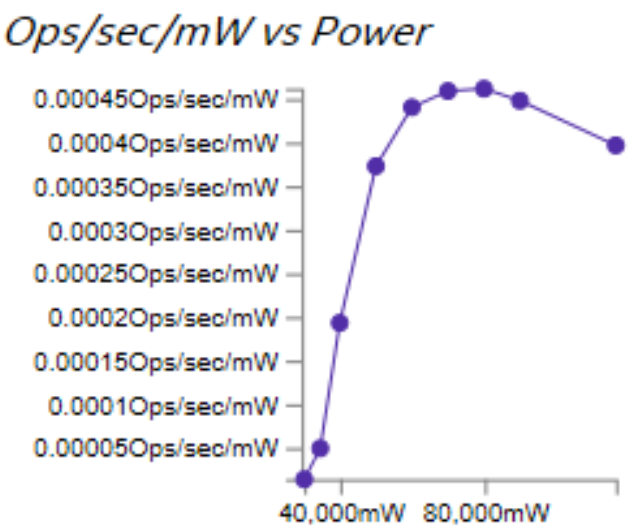
CPU/GPU レンダリングの調整 - GPU レンダリング解析 (Linux*、Android* のみ)

- OpenGL* ES アプリケーションのパフォーマンス・クリティカルな API 呼び出しを検出
- システムのボトルネックである仮想 Xen* ハイパーバイザー・ドメインを特定

CPU のロードバランスの調整 - CPU 解析

- パフォーマンスが向上するように CPU のロードバランスを調整
- メモリー転送時間、CPU バランス、CPU コンテキスト・スイッチ、FPU 使用率、ワークロード待機時間などを評価

電力解析により電力/熱制限を満たし、バッテリー寿命を最適化



電力解析をワークフローに統合

- Eclipse* プラグインにより素早く簡単に解析結果を取得
- インテル® SoC Watch はライトウェイトのコマンドライン・インターフェイスで広範なデータを提供
- インテル® VTune™ Amplifier にデータをインポートして、タイムライン解析を実行したり、パフォーマンス・データと関連付けることが可能

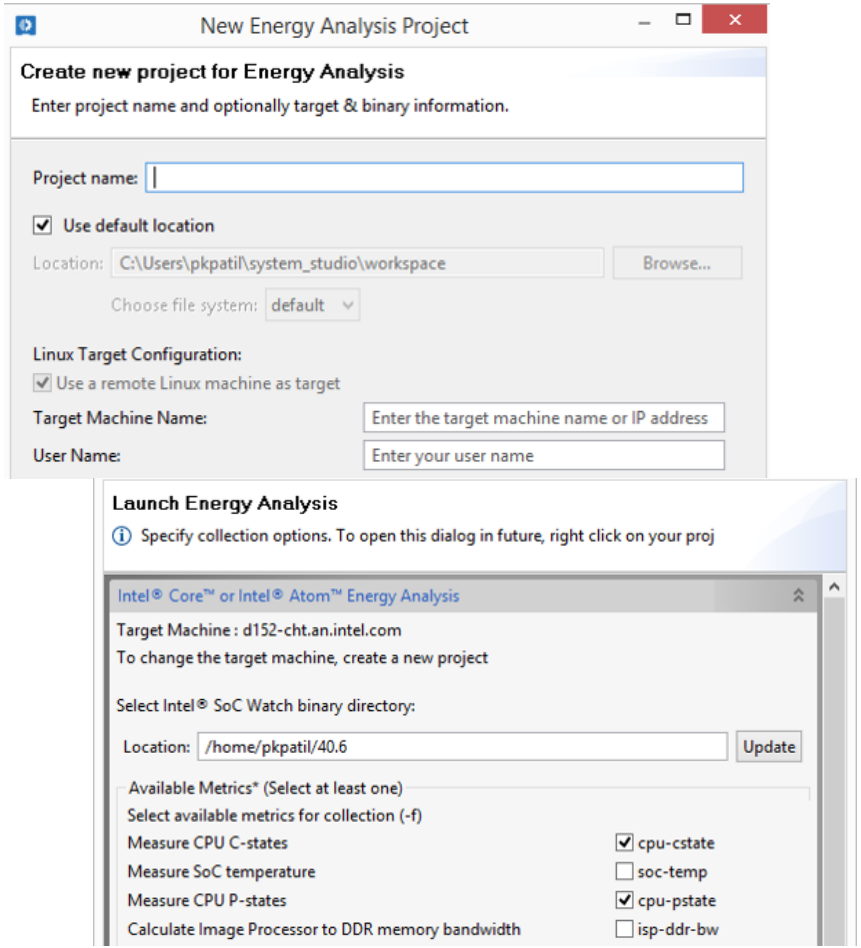
バージョン 2019 の新機能 (ハイライト)

- Linux* ターゲットの “自動インストール” がさらに簡単に
- Android* ターゲットのサポートを追加

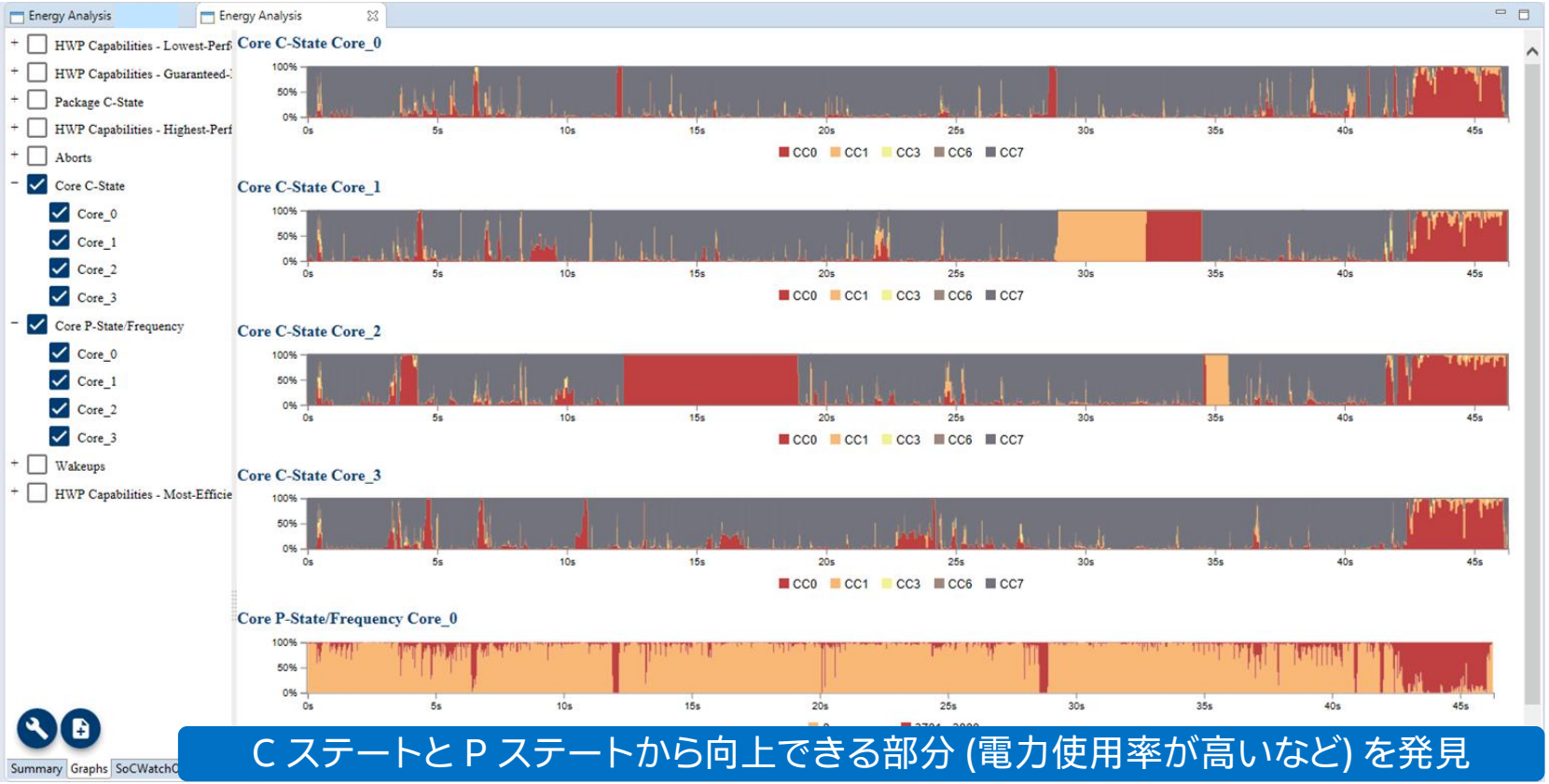
簡単な電力解析

Eclipse* 向け電力解析プラグイン

- 電力データの収集が容易に
- Windows* または Linux* ホストから Linux* ターゲット
- ドライバーを作成して挿入
- 利用可能なメトリックを動的に取得
- チェックボックスで収集オプションを選択



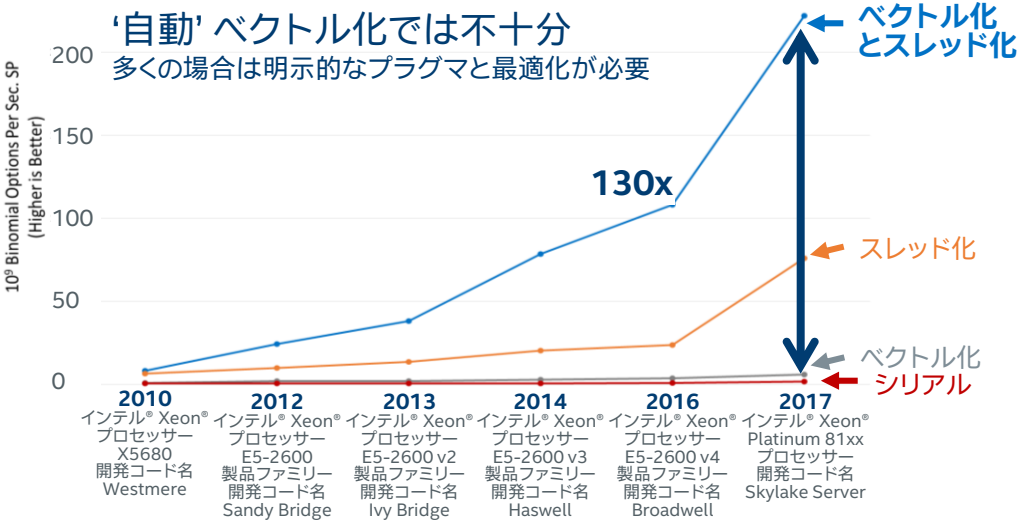
詳細な電力解析のタイムライン



C ステートと P ステートから向上できる部分 (電力使用率が高いなど) を発見

インテル® Advisor によるベクトル化とスレッド化の最適化

新しいハードウェア世代ごとにパフォーマンスが向上



パフォーマンスの良い現代化されたコード

- インテル® アドバンスト・ベクトル・エクステンション (インテル® AVX-512 および インテル® AVX) 向けにベクトル化
- 効率良いメモリーアクセス
- スレッド化

機能

- ベクトル化とその最適化
- メモリーパターン解析
- 迅速なスレッドのプロトタイプ生成

ベンチマーク: 2 項オプション価格モデル software.intel.com/en-us/articles/binomial-options-pricing-model-code-for-intel-xeon-phi-coprocessor (英語)

パフォーマンス結果は 2017 年 8 月時点のテスト結果に基づいたものであり、公開されている利用可能なすべてのセキュリティー・アップデートが適用されていない可能性があります。詳細については、構成の開示を参照してください。絶対的なセキュリティーを提供できる製品はありません。性能やベンチマーク結果について、さらに詳しい情報をお知りになりたい場合は、http://www.intel.com/benchmarks/ (英語) を参照してください。2010 - 2017 のシステム構成については、補足資料の「ベクトル化およびスレッド化するか、パフォーマンスを諦めるか」にあるベンチマークを参照してください。2017 年 8 月時点のインテルによるテスト。

詳細: isus.jp/intel-advisor-xe/

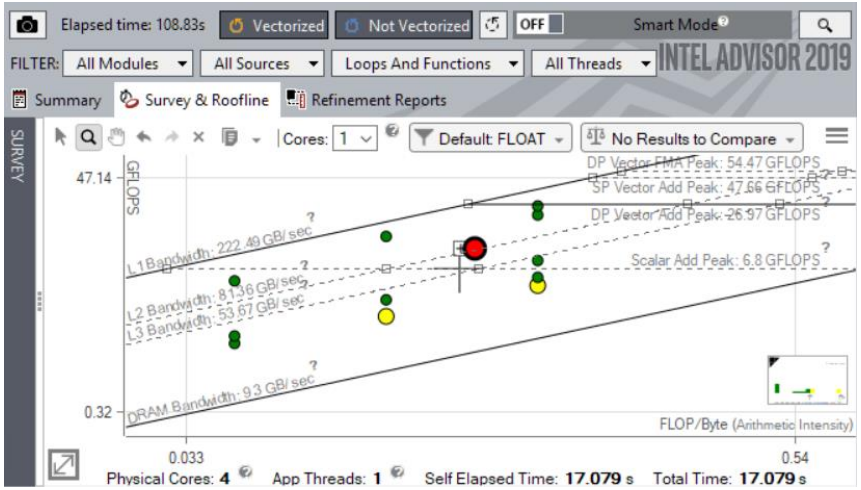
XLSOFT
インテル® コンパイラーでは、インテル® マイクロプロセッサーに限定されない最適化に関して、他社製マイクロプロセッサー用に同等の最適化を行えないことがあります。これには、インテル® ストリーミング SIMD 拡張命令 2、インテル® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサーに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサー依存の最適化は、インテル® マイクロプロセッサーでの使用を前提としています。インテル® マイクロアーキテクチャーに限定されない最適化のなかにも、インテル® マイクロプロセッサー用のものがあります。この注意事項で言及した命令セットの詳細については、該当する製品のユーザー・リファレンス・ガイドを参照してください。注意事項の改訂 #20110804

効率的な最適化手法の発見

インテル® Advisor - キャッシュを考慮したルーフライン解析

ルーフライン・パフォーマンスの詳細な情報

- パフォーマンスの悪いループをハイライト
- 各ループの「最大」パフォーマンスを表示
  - 最適化可能なループ
  - 最適化する価値のあるループ
- ボトルネックの原因の可能性を表示
- 次の最適化ステップを提案



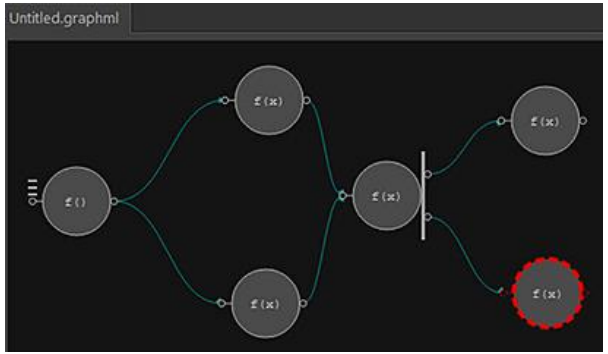
「インテル® Advisor の新しい “統合ルーフライン” は素晴らしい機能です。メモリー転送の最適化とベクトル化という重要で難しい課題に、ステップ・バイ・ステップのアプローチで対処できるのです。」

Onera (フランス国立航空宇宙研究所)
ソフトウェア・アーキテクト、Nicolas Alferez 氏

並列処理を視覚化

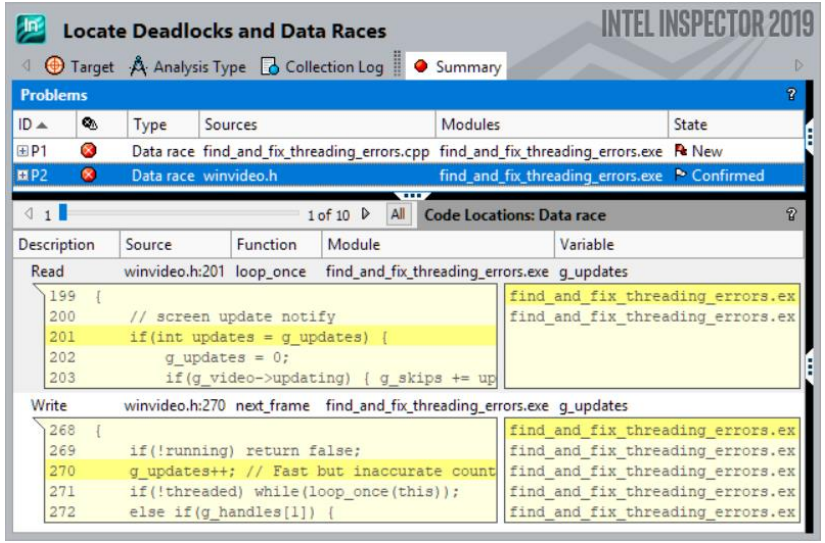
インテル® Advisor - フローグラフ・アナライザー (FGA)

- フローグラフ・アナライザー・モジュールによる
  インタラクティブなビルド、検証、アルゴリズム解析
  - コードスタブおよびインテル® TBB フローグラフを視覚的に生成して、並列 C++ プログラムの生成を開始
  - アルゴリズムのノードやエッジをクリックしてズームし、並列データとプログラムフローを理解
  - FGA ダッシュボードでアルゴリズムのロードバランス、コンカレンシー、その他の並列特性を解析し、プログラムを細かくチューニング
- インテル® TBB フローグラフの完全サポート、OpenMP* 5.0 (ドラフト) OMPT API の初期サポート



インテル® Inspector によるメモリーとスレッドのデバッグ

メモリーリーク、異常、データ競合、デッドロックの特定とデバッグ



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て済む
- 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

バージョン 2019 の新機能

不揮発性メモリーエラーを発見

- 不足している/冗長なキャッシュフラッシュ
- ストアフェンスの不足
- アウトオブオーダーの不揮発性メモリーストア
- PMDK トランザクションの Redo (やり直し) ログエラー

新しい情報: インテル® Inspector - 永続性インスペクターの使用法

詳細: isus.jp/intel-inspector-xe/

[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

競合状態は診断が困難

常に発生しないため簡単に再現できない - インテル® Inspector

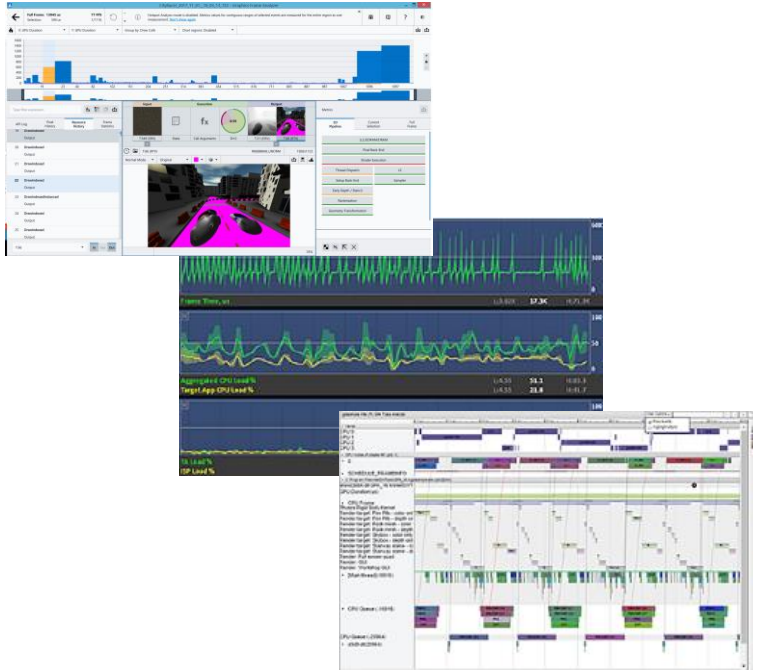
正しい結果

スレッド 1	スレッド 2		共有カウンター
			0
読み取りカウント		←	0
インクリメント			0
書き込みカウント		→	1
	読み取りカウント	←	1
	インクリメント		1
	書き込みカウント	→	2

間違った結果

スレッド 1	スレッド 2		共有カウンター
			0
読み取りカウント		←	0
	読み取りカウント	←	0
インクリメント			0
	インクリメント		0
書き込みカウント		→	1
	書き込みカウント	→	1

インテル® Graphics Performance Analyzer で グラフィックス・アプリケーションを解析、最適化、デバッグ



ゲームの最適化とパフォーマンスの向上

- リアルタイム・ハードウェア・メトリックの詳細な解析
- 詳細なフレーム解析により FPS を向上
- 困難な CPU/GPU の問題を特定
- 複数のプラットフォームと GPU をサポート
- ソースコードや外部の変更は不要

バージョン 2019 の新機能

- インテル® NUC キット NUC8I7HVK の AMD Radeon* RX Vega-M グラフィックス・カードのメトリックのサポート
- 描画呼び出し内の問題があるテクスチャーの識別、2x2 テクスチャーに対する新しい試行によるパフォーマンス・デルタの解析
- DirectX* 11 フレームの複数の選択された描画呼び出しのジオメトリーの表示および最適化

詳細: isus.jp/intel-system-studio/

インテル® System Studio コンポーネント・ツールと機能

デバッグ

インテル® System Debugger
インテル® Debug Extensions for WinDbg

解析

インテル® VTune™ Amplifier
電力解析
インテル® Advisor
インテル® Inspector
インテル® Graphics Performance Analyzers

ビルド

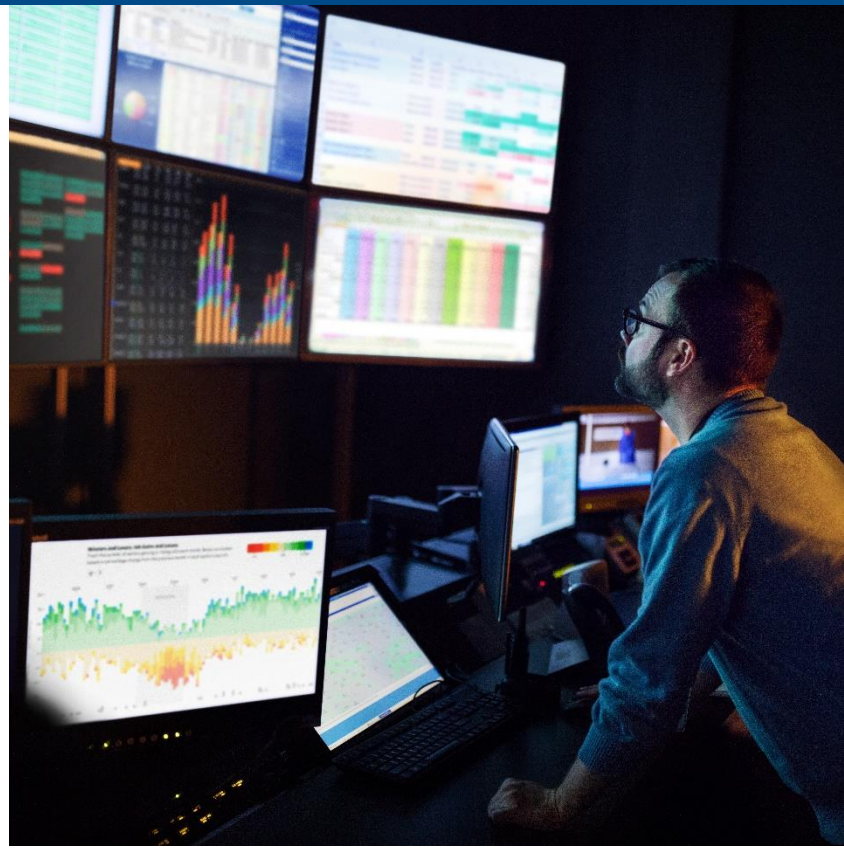
インテル® C++ コンパイラー
インテル® MKL
インテル® DAAL
インテル® IPP
インテル® TBB
OpenCL* ツール
IoT 接続ツール

インテル® コンパイラーによる高速でスケーラブルな並列コード

最先端の C/C++ コード・パフォーマンスを実現、最新のインテル® プロセッサーの能力を最大限に活用

- Intel Atom® プロセッサー、インテル® Core™ プロセッサー、インテル® Xeon® プロセッサーを含むインテル® アーキテクチャー向けに最適化およびベクトル化されたコードを開発
- 優れた並列パフォーマンスを実現 – (OpenMP* を使用した) ベクトル化とスレッド化により、インテル® アドバンスト・ベクトル・エクステンション 512 (インテル® AVX-512) 命令を含む最新の SIMD 対応ハードウェアを最大限に活用
- 最新の言語標準と OpenMP* 標準規格の利用、主要なコンパイラーおよび IDE との互換性

詳細: isus.jp/c-compilers/



コンパイラーはインテル® System Studio の主要コンポーネント

ソフトウェア開発者にとって重要なポイント		課題	インテル® C/C++ コンパイラー
	効率とパフォーマンス	さまざまなプラットフォームや世代向けの最適化よりも、製品の開発に集中したい	ベクトル命令を利用し、パフォーマンス向上のためコードを調整して、ターゲット・プロセッサー向けの強力なチューニングが可能
	開発期間の短縮	主要な組込み/IoT オペレーティング・システムと言語がサポートされていなければならない	広範なターゲットにわたって最先端の C++ 機能と主要なコンパイラーおよび IDE との互換性を提供
	システムの信頼性	発見が困難なシステムやメモリー/スレッドの問題を切り分け、デバッグしなければならない	インテルのデバッグおよび解析ツールと連携して、システムの信頼性に関する問題を排除

インテル® コンパイラー 19.0 の新機能

C++17 標準機能の追加サポート

- ラムダ式と定数式のサポートを向上
- GNU* C++ コンパイラーおよび Microsoft* コンパイラーとの互換性を向上

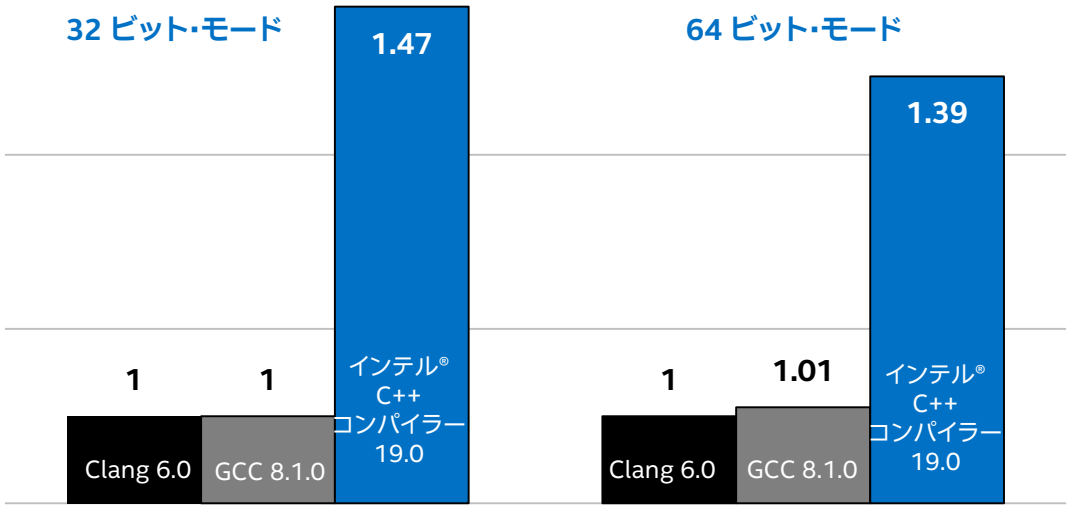
標準規格に基づく並列化により C++ 開発者を支援

- OpenMP* 5.0[1] TR7 の一部をサポート
- 最新の並列化仕様によりコードを現代化

[1]OpenMP* 5.0 はドラフト

インテル® Core™ プロセッサーにおいて組込みアプリケーションのパフォーマンスを大幅に向上 - インテル® C++ コンパイラー

インテル® Core™ i7 プロセッサー・ファミリー上での Coremark*-Pro ベンチマーク



パフォーマンス結果は 2018 年 8 月 26 日時点のテスト結果に基づいたものであり、公開されている利用可能なすべてのセキュリティー・アップデートが適用されていない可能性があります。詳細については、構成の開示を参照してください。絶対的なセキュリティーを提供できる製品はありません。性能に関するテストに使用されるソフトウェアとワークロードは、性能がインテル® マイクロプロセッサー用に最適化されていることがあります。SYSmark* や MobileMark* などの性能テストは、特定のコンピューター・システム、コンポーネント、ソフトウェア、操作、機能に基づいて行ったものです。結果はこれらの要因によって異なります。製品の購入を検討される場合は、他の製品と組み合わせた場合の本製品の性能など、ほかの情報や性能テストも参考にして、パフォーマンスを総合的に評価することをお勧めします。詳細については、パフォーマンス・ベンチマーク・テストの開示 (英語) を参照してください。
2018 年 8 月 26 日時点のインテルによるテスト。システム構成: インテル® Core™ i7-8700K プロセッサー @ 3.70GHz、64GB RAM、ハイパースレッディング有効。ソフトウェア: インテル® コンパイラー 19.0、GCC 8.1.0、PGI* 18.5、Clang/LLVM 6.0。Linux* OS: Red Hat* Enterprise Linux* Server 7.4 (Maipo)、3.10.0-693.el7.x86_64。Coremark*-Pro ベンチマーク (www.eembc.org (英語))。C++ テストには SmartHeap* 10 を使用。
インテル® C++ コンパイラー 19.0: -O3 -no-prec-div -ipo -xCORE-AVX2。GCC 8.1.0: -m32 -Ofast -mfpmath=sse -flto -march=native -funroll-loops -ffat-lto-objects。Clang 6.0: -m32 -Ofast -mfpmath=sse -flto -march=native -funroll-loops。

インテル® コンパイラーでは、インテル® マイクロプロセッサーに限定されない最適化に関して、他社製マイクロプロセッサー用に同等の最適化を行えないことがあります。これには、インテル® ストリーミング SIMD 拡張命令 2、インテル® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサーに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサー依存の最適化は、インテル® マイクロプロセッサーでの使用を前提としています。インテル® マイクロアーキテクチャーに限定されない最適化のなかにも、インテル® マイクロプロセッサー用のものがあります。この注意事項で言及した命令セットの詳細については、該当する製品のユーザー・リファレンス・ガイドを参照してください。注意事項の改訂 #20110804

インテル® MKL による高速でスケーラブルなコード

- 高度に最適化、スレッド化、ベクトル化された数学関数により、マシンラーニング・アプリケーションにおける計算を高速化
- 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、スプラインなどの主な機能を提供
- コードを分岐せずに各プロセッサー向けに最適化されたコードを自動ディスパッチ
- シングルコアのベクトル化およびキャッシュ効率を向上する最適化
- 最小限の労力で優れたパフォーマンスを達成

インテル® MKL に含まれる機能

- 密/スパース線形代数
- 高速フーリエ変換
- ベクトル演算
- ベクトル RNG
- 高速ポアソンソルバー
- その他

インテル® MKL 2019 の新機能

JIT 高速小行列乗算

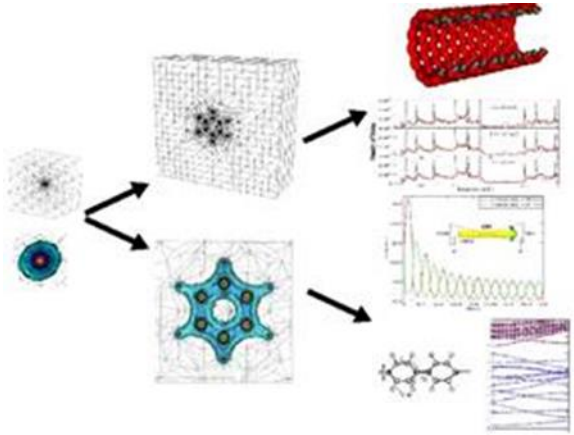
- JIT 機能によりインテル® アドバンスト・ベクトル・エクステンション (インテル® AVX-512 およびインテル® AVX2) 向け S/DGEMM の速度を向上

スパース QR ソルバー

- スパース線形方程式、スパース線形最小 2 乗問題、固有値問題、ランクと null 空間の決定などのソルバー

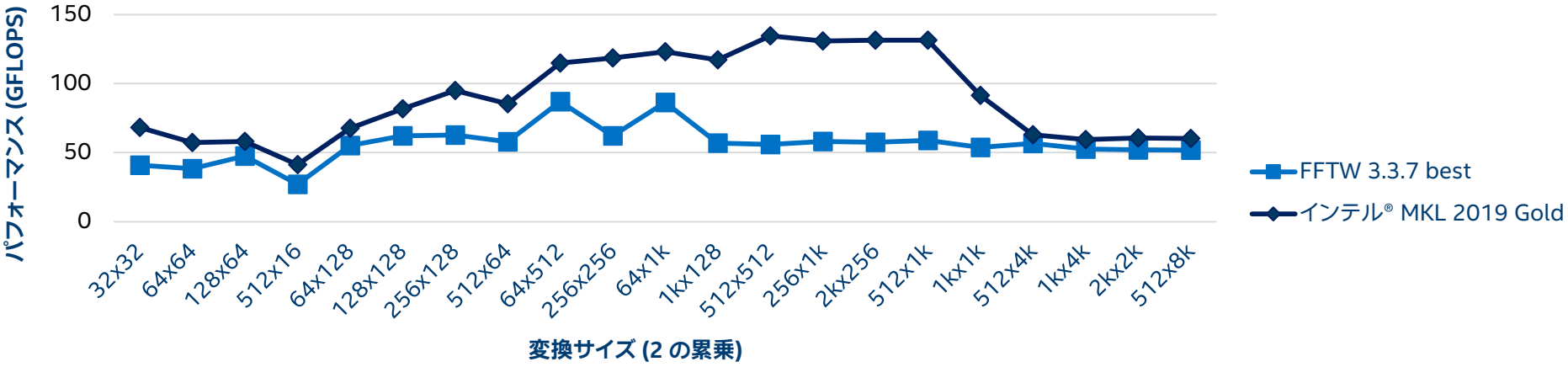
多項式向け乱数生成

- 金融、地質学、生物学分野向けに高度に最適化された多項式乱数ジェネレーター



インテル® Core™ プロセッサーにおいて FFT パフォーマンスを大幅に向上 - インテル® MKL

インテル® MKL 2019 Gold と FFTW* の 2D FFT パフォーマンスの比較
単精度 2D FFT - インテル® Core™ i5-7600 プロセッサー @ 3.50GHz



パフォーマンス結果は 2018 年 7 月 12 日時点のテスト結果に基づいたものであり、公開されている利用可能なすべてのセキュリティー・アップデートが適用されていない可能性があります。詳細については、構成の開示を参照してください。絶対的なセキュリティーを提供できる製品はありません。
性能に関するテストに使用されるソフトウェアとワークロードは、性能がインテル® マイクロプロセッサー用に最適化されていることがあります。SYSmark* や MobileMark* などの性能テストは、特定のコンピューター・システム、コンポーネント、ソフトウェア、操作、機能に基づいて行ったものです。結果はこれらの要因によって異なります。製品の購入を検討される場合は、他の製品と組み合わせた場合の本製品の性能など、ほかの情報や性能テストも参考にして、パフォーマンスを総合的に評価することをお勧めします。詳細については、パフォーマンス・ベンチマーク・テストの開示 (英語) を参照してください。
2018 年 7 月 12 日時点のインテルによるテスト。システム構成: インテル® Core™ i5-7600 プロセッサー、4 コア、3.50GHz、6MB CPU キャッシュ、64GB RAM。Linux* OS: Red Hat* Enterprise Linux* 7.2。

XLSOFT
インテル® コンパイラーでは、インテル® マイクロプロセッサーに限定されない最適化に関して、他社製マイクロプロセッサー用に同等の最適化を行えないことがあります。これには、インテル® ストリーミング SIMD 拡張命令 2、インテル® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサーに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサー依存の最適化は、インテル® マイクロプロセッサーでの使用を前提としています。インテル® マイクロアーキテクチャーに限定されない最適化のなかにも、インテル® マイクロプロセッサー用のものがあります。この注意事項で言及した命令セットの詳細については、該当する製品のユーザー・リファレンス・ガイドを参照してください。注意事項の改訂 #20110804

インテル® DAAL によるエッジでの解析のスピードアップ

マシンラーニングおよびビッグデータ解析のパフォーマンスを高速化

- アプリケーションがより速くより優れた予測を行えるように支援
- 最高のパフォーマンスを引き出せるようにデータの取り込みとアルゴリズムの計算を最適化
- 広範なアプリケーションのニーズを満たす、オフライン、ストリーミング、分散型使用モデルをサポート
- エッジデバイスとクラウドサービス間で解析ワークロードを分割してアプリケーション全体のスループットを最適化

詳細: isus.jp/intel-daal/

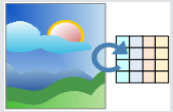
バージョン 2019 の新機能

新しいアルゴリズム

- **ハイパフォーマンス・ロジスティック回帰**、最も広く利用されている分類アルゴリズム
- **拡張勾配ブースティング機能**、ユーザー定義のコールバックによる不正確な分割計算およびアルゴリズムレベルの計算のキャンセルにより柔軟性を拡張
- **ユーザー定義の CSV/IDBC データソースのデータ変更プロシージャー**、広範な特徴抽出および変換手法を実装

前処理 → 変換 → 解析 → モデリング → 検証 → 意思決定

展開、
フィルタリング、
正規化

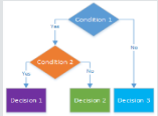


集計、
次元縮小

サマリー統計、
クラスタリングなど

マシンラーニング (訓練)、
パラメーター推定、
シミュレーション

仮説検証、
モデルエラー



予測、
決定木など

データ変換および解析アルゴリズム

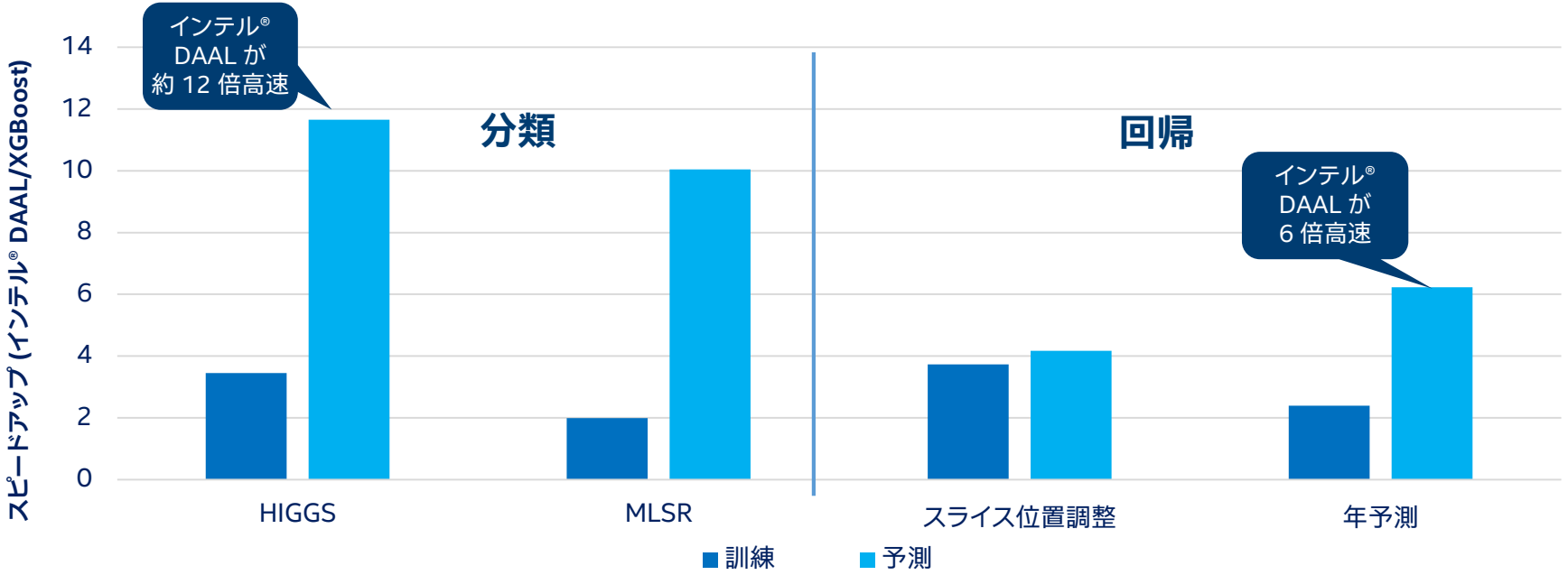
インテル® DAAL

- データセットの基礎統計
  - 低次モーメント
  - 四分位数
  - 順序統計量
- 相関と依存関係
  - コサイン距離
  - 相関距離
  - 分散共分散行列
- 行列の因数分解
  - SVD
  - QR
  - コレスキー
- 次元縮小
  - PCA
  - 相関ルールマイニング(アプリオリ)
  - 最適化ソルバー(SGD、AdaGrad、lBFGS)
- 外れ値検出
  - 単変量
  - 多変量
- 数学関数(exp、log、…)

バッチ処理をサポートするアルゴリズム

バッチ、オンライン、分散処理をサポートするアルゴリズム

インテル® DAAL 2019 と XGBoost のパフォーマンスの比較



XGBoost オープンソース・プロジェクト
パフォーマンス結果は 2018 年 7 月 9 日時点のテスト結果に基づいたものであり、公開されている利用可能なすべてのセキュリティー・アップデートが適用されていない可能性があります。詳細については、構成の開示を参照してください。絶対的なセキュリティーを提供できる製品はありません。
性能に関するテストに使用されるソフトウェアとワークロードは、性能がインテル® マイクロプロセッサー用に最適化されていることがあります。SYSmark* や MobileMark* などの性能テストは、特定のコンピューター・システム、コンポーネント、ソフトウェア、操作、機能に基づいて行ったものです。結果はこれらの要因によって異なります。製品の購入を検討される場合は、他の製品と組み合わせた場合の本製品の性能など、ほかの情報や性能テストも参考にして、パフォーマンスを総合的に評価することをお勧めします。詳細については、パフォーマンス・ベンチマーク・テストの開示 (英語) を参照してください。2018 年 7 月 9 日時点のインテルによるテスト。システム構成: インテル® Xeon® Platinum 8180 プロセッサー、2x28 コア、2.50GHz、192GB、12x16GB DDR4-2666。インテル® DAAL 2019。Red Hat* Enterprise Linux* 7.2。

XLSOFT
インテル® コンパイラーでは、インテル® マイクロプロセッサーに限定されない最適化に関して、他社製マイクロプロセッサー用に同等の最適化を行えないことがあります。これには、インテル® ストリーミング SIMD 拡張命令 2、インテル® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサーに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサー依存の最適化は、インテル® マイクロプロセッサーでの使用を前提としています。インテル® マイクロアーキテクチャーに限定されない最適化のなかにも、インテル® マイクロプロセッサー用のものがあります。この注意事項で言及した命令セットの詳細については、該当する製品のユーザー・リファレンス・ガイドを参照してください。注意事項の改訂 #20110804

インテル® IPP により画像、ビジョン、信号、セキュリティー、ストレージ・アプリケーションのパフォーマンスを向上

画像、信号、データ処理および暗号化計算を高速化

- マルチコア、マルチ OS、マルチプラットフォームに対応した計算集約型の高度に最適化された関数
- ハイパフォーマンスで、使いやすい、プロダクション環境対応の API により素早くアプリケーション・パフォーマンスを向上
- ソフトウェアの開発と保守にかかる費用を軽減し、開発期間を短縮

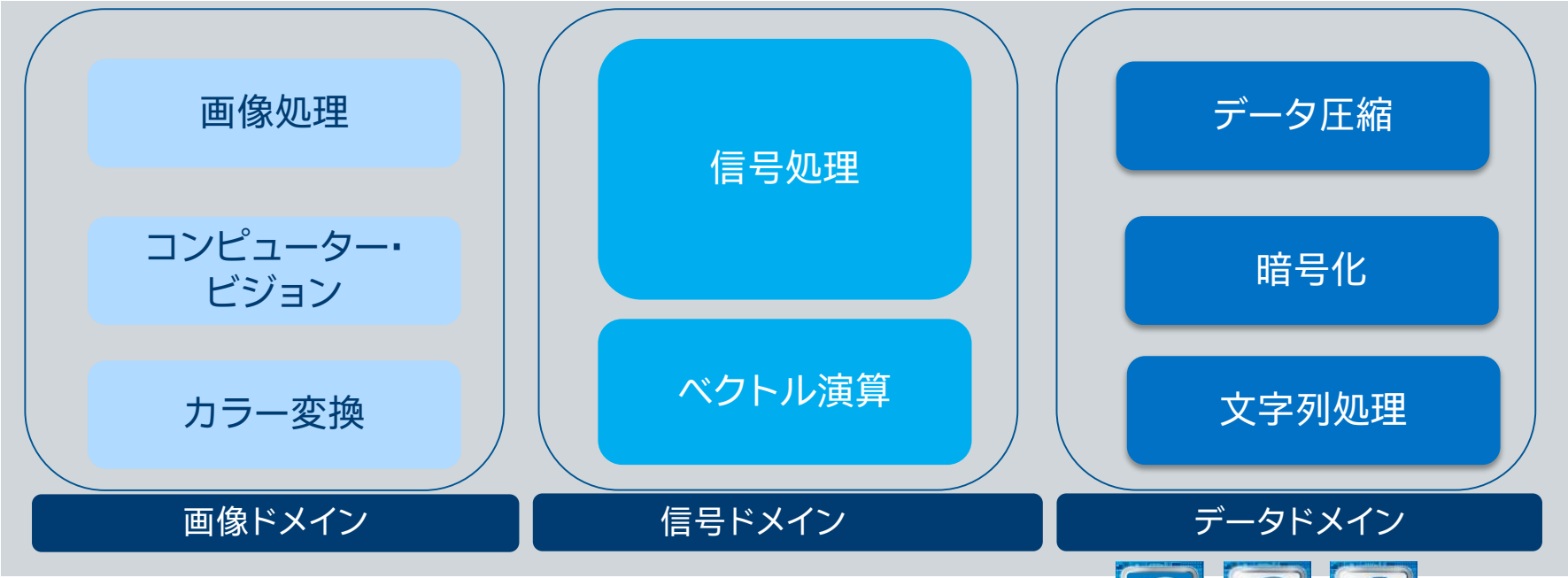
バージョン 2019 の新機能

- エントロピーが高いデータの LZ4 圧縮/展開パフォーマンスを向上
- RBG イメージから CIE Lab 色モデル (およびその逆) に変換する新しい色変換関数
- インテル® IPP Cryptography ライブラリーのオープンソース・ディストリビューション
- インテル® AVX-512 (英語) およびインテル® AVX2 (英語) 命令セット向けの最適化を拡張
- 各種画像処理関数に OpenMP* およびインテル® TBB のスレッド層のサポートを追加

詳細: isus.jp/intel-ipp/

インテル® IPP

ハイパフォーマンスで、使いやすい、プロダクション環境対応の API

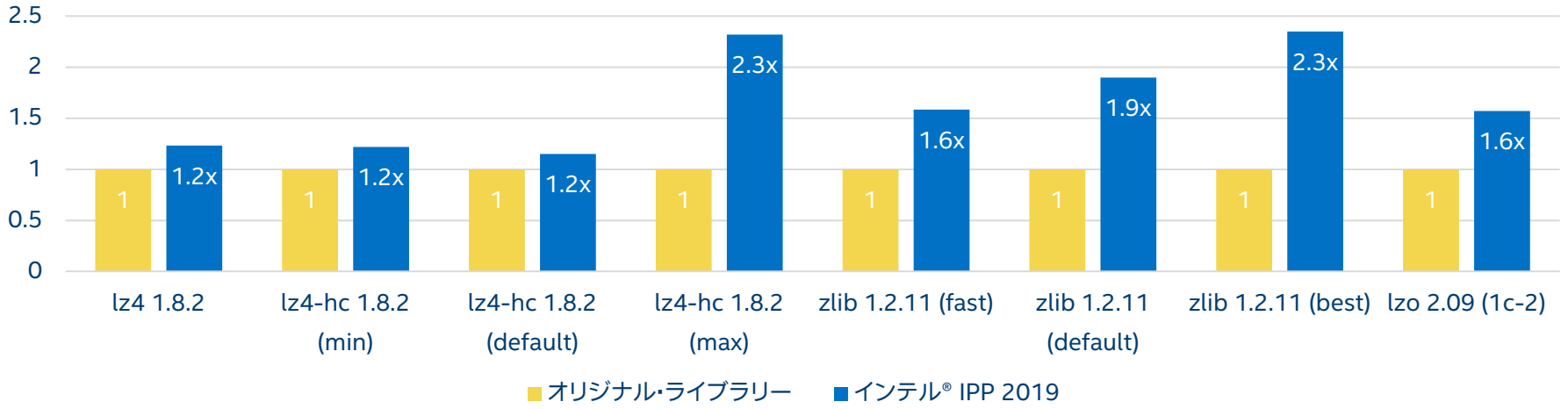


インテル® アーキテクチャー・ベースのプラットフォーム

データ圧縮のパフォーマンスを向上

インテル® IPP

インテル® IPP 2019 と LZ4、Zlib、LZO ライブラリーの データ圧縮パフォーマンスの比較



パフォーマンス結果は 2018 年 8 月 15 日時点のテスト結果に基づいたものであり、公開されている利用可能なすべてのセキュリティー・アップデートが適用されていない可能性があります。詳細については、構成の開示を参照してください。絶対的なセキュリティーを提供できる製品はありません。
性能に関するテストに使用されるソフトウェアとワークロードは、性能がインテル® マイクロプロセッサー用に最適化されていることがあります。SYSmark* や MobileMark* などの性能テストは、特定のコンピューター・システム、コンポーネント、ソフトウェア、操作、機能に基づいて行ったものです。結果はこれらの要因によって異なります。製品の購入を検討される場合は、他の製品と組み合わせた場合の本製品の性能など、ほかの情報や性能テストも参考にして、パフォーマンスを総合的に評価することをお勧めします。詳細については、パフォーマンス・ベンチマーク・テストの開示 (英語) を参照してください。
2018 年 8 月 15 日時点のインテルによるテスト。システム構成: ハードウェア: インテル® Core™ i5-7600 プロセッサー @ 3.50GHz、4 コア、ハイパースレッディング無効。キャッシュ: L1=32KB、L2=256KB、L3=6MB。メモリー: 64GB。Linux* OS: Red Hat* Enterprise Linux* Server 7.2。

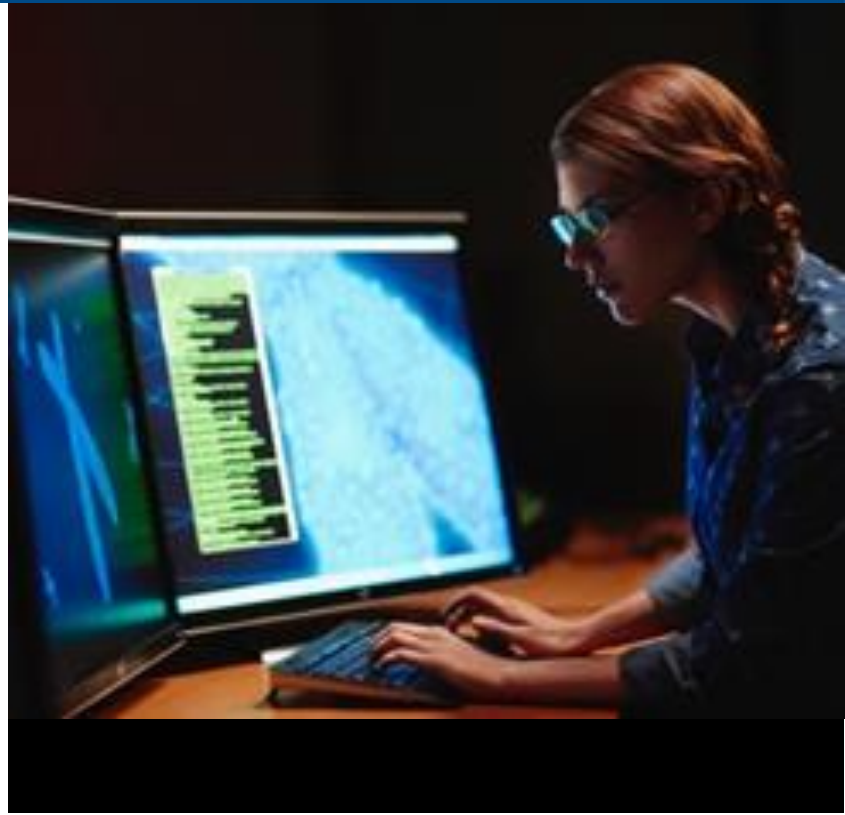
XLSOFT
インテル® コンパイラーでは、インテル® マイクロプロセッサーに限定されない最適化に関して、他社製マイクロプロセッサー用に同等の最適化を行えないことがあります。これには、インテル® ストリーミング SIMD 拡張命令 2、インテル® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサーに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサー依存の最適化は、インテル® マイクロプロセッサーでの使用を前提としています。インテル® マイクロアーキテクチャーに限定されない最適化のなかにも、インテル® マイクロプロセッサー用のものがあります。この注意事項で言及した命令セットの詳細については、該当する製品のユーザー・リファレンス・ガイドを参照してください。注意事項の改訂 #20110804

インテル® TBB による高度なスレッド化

スレッド化手法を利用して C++ でマルチコア・パフォーマンスとヘテロジニアス・コンピューティングを活用

- CPU/GPU にわたって計算負荷の高い作業を並列処理 — C++ で高レベルのシンプルなソリューションを提供
- 並列プログラミング向けの最も機能が豊富で包括的なソリューション
- 優れた移植性、構成の容易性、コスト、アプローチを備えた将来にわたるスケーラビリティー

詳細: isus.jp/intel-tbb/



インテル® TBB 2019 の新機能

言語サポートの拡張

- Python* で記述された数値演算コードを処理
- C++17 Parallel STL、OpenCL*、Python* conda 言語をサポート

ブロック化された並列処理を最適化

- 新しいテンプレートにより汎用 C++11 多次元配列を最適化

並列性とヘテロジニアスへの対応が向上

- 改良されたタスク・スケジューラー/アナライザー、OpenCL* デバイス選択、汎用並列アルゴリズム、ユーザー・イベント・トレース API によりインテル® TBB のフローグラフとインテル® VTune™ Amplifier の機能を拡張
- 機能ノードの lightweight ポリシーによりフローグラフのタスク・スケジューリング・オーバーヘッドを軽減
- OpenCL* プラットフォームのサポートにデフォルトデバイスのオプションを追加

開発用 OS を選択可能

- Linux*、Windows*、Android*、macOS* のサポートを拡張

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。

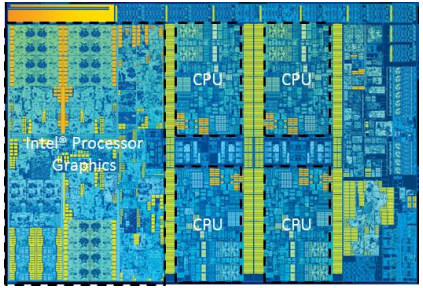
インテル® TBB のスレッド化モデルを利用する利点

- スレッドを操作する代わりに**タスクを指定**。論理タスクをスレッドにマップ (入れ子構造の並列処理を完全サポート)。
- **実証済みの効率良い並列処理パターン**
- ワークスチールを使用して実行時間が不明なタスクのロードバランスをサポートし、**低オーバーヘッドの**ポリモーフィズムを活用
- **フローグラフ機能**により依存性とデータフロー・グラフを簡単に表現
- **高レベルの並列アルゴリズム**、コンカレント・コンテナー、低レベルのビルディング・ブロック (スケーラブル・メモリー・アロケーター、ロック、アトミック操作など) を提供

ヘテロジニアス・サポート

インテル® TBB

ヘテロジニアス調整レイヤーとしてインテル® TBB のフローグラフは最適化の可能性を保持し、既存のモデルで構成



CPU、統合 GPU など

+

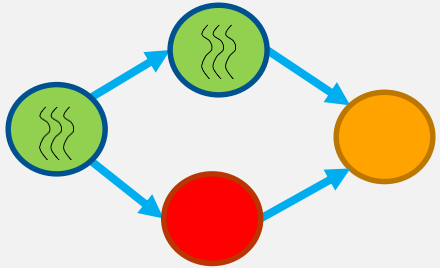
インテル® TBB
OpenVX*
OpenCL*
COI/SCIF
…

ライブラリー実装の**構成レイヤー**としてのインテル® TBB

- 1 つのスレッドエンジンで CPU 側の全処理に対応

調整レイヤーとしてのインテル® TBB のフローグラフ

- ヘテロジニアス・ハードウェアとソフトウェアをつなぎ合わせる
- ブロック間の並列性を示し、統合を容易にする



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。
OpenVX および OpenVX ロゴは Khronos の商標です。

新しいツール

ヘテロジニアス計算アプリケーションにおけるインテルの強み

OpenCL* ツール

カスタムコードのビルド、計算パフォーマンスと電力の最適化

- 使いやすい、カーネルビルド、解析、デバッグツール
- 柔軟にコードをカスタマイズして、インテル® プロセッサー、GPU (インテル® プロセッサー・グラフィックス) 向けに ‘独自の’ アルゴリズムを開発
- 負荷が高い計算をインテル® プロセッサー・グラフィックスにオフロード
- ハイパフォーマンスな画像/ビデオ処理を実現
- 改良されたカーネル開発フレームワークにより生産性を向上
- 複数のインテル® プラットフォームで実行するエンドツーエンドのソリューションスタックを開発する際の総所有コストを削減

詳細: isus.jp/intel-opencl/

対象アプリケーションの種類

システム、組込み/IoT、ビジュアル・コンピューティング

ユーザー

ソリューション・アーキテクト、アルゴリズム開発者 (R&D)、アプリケーション開発者、ハードウェア/ソフトウェア・コンポーネント・ビルダー、リセラー/サービス・プロバイダー (OEM、CSP など)

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。

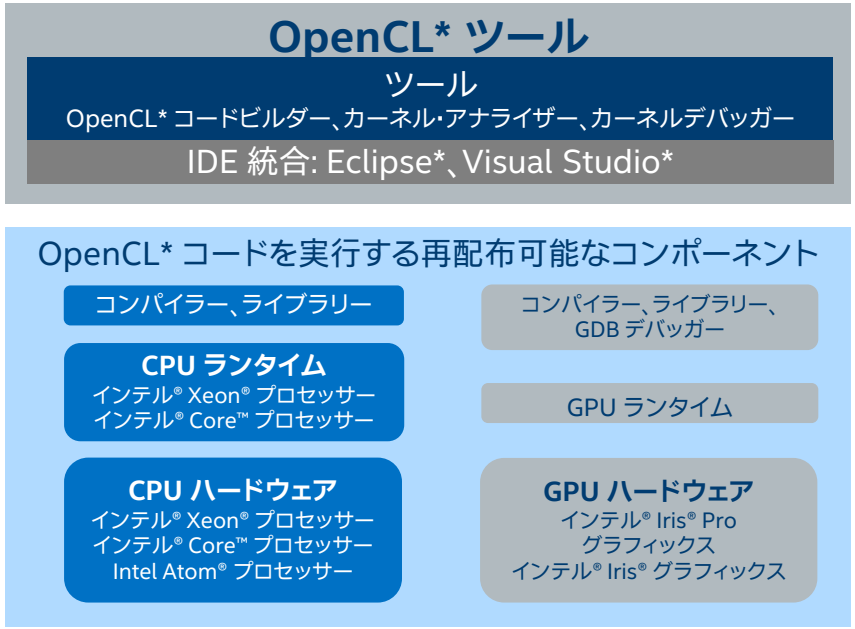
OpenCL* ツール

OpenCL* プログラミングの包括的な環境

- Eclipse* ベースの IDE (Linux*)
- Microsoft* Visual Studio* に対応 (Windows*)
- OpenCL* キーワードのハイライト
- カーネル解析ツールおよびレポート表示

共通の環境からホスト/アクセラレーター・コードを最適化

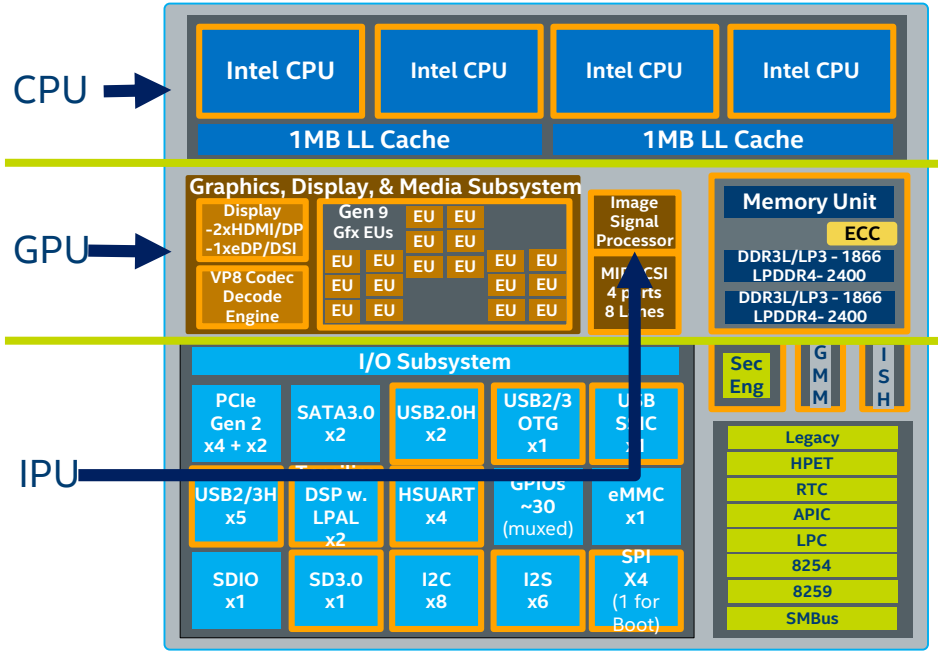
- ホストのイベントとキューを追跡
- CPU/GPU のシステム・パフォーマンスを最適化



インストールの詳細は入門ガイド (英語) を参照

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。

OpenCL* ツールの利点: インテル® プラットフォームのハードウェア機能にアクセス



例

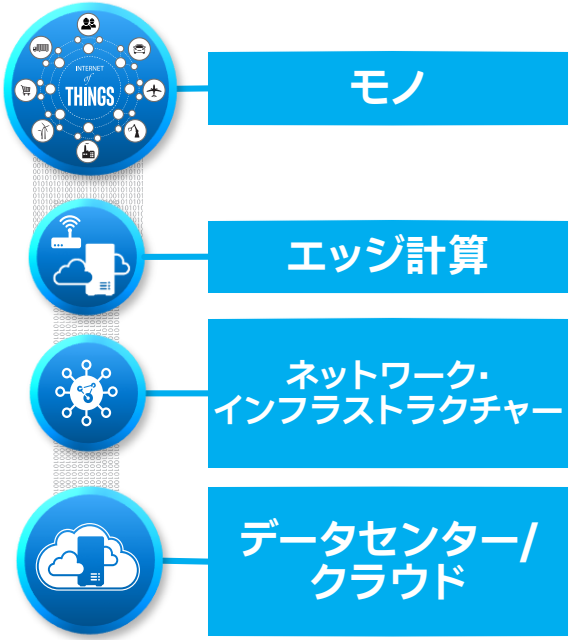
カスタム･アルゴリズム	最適化されたライブラリー
インテル® C/C++ コンパイラー	▪ インテル® MKL ▪ インテル® IPP ▪ インテル® TBB
OpenCL*	インテル® TBB、推論エンジン、インテル® OpenVINO™ ツールキットの OpenVX*、インテル® Media Server Studio
N/A (ハードウェアは カスタム開発に 適していない)	OpenVX*

OpenCL* は開発者がインテル® プロセッサー･グラフィックス/GPU 向けの ‘カスタム･アルゴリズム’ を記述する唯一の方法

XLSOFT
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。
OpenVX および OpenVX ロゴは Khronos の商標です。

IoT 接続ツールを利用したソリューションの開発

IoT 接続ツールはすべての IoT ソリューション開発者にとって最も重要なライブラリーのコレクション
IDE インターフェイスによる統合とプロジェクト･テンプレートを利用可能



- **標準化されたオープンソースの抽象化ライブラリーとツール**を提供
- 400 を超える**センサー･ライブラリー**を含む

MRAA　UPM

- 高度な**クラウドコネクター**を含む

IBM BlueMix　Microsoft Azure　amazon web services™　Google Cloud Platform

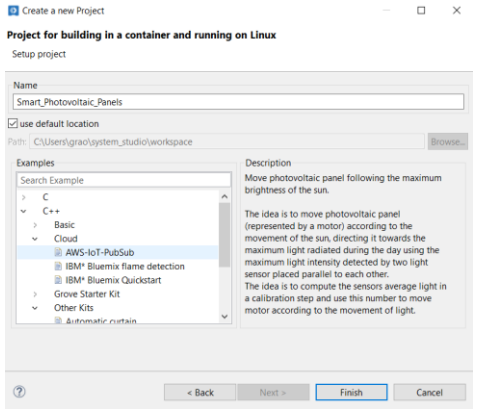
新しいクラウド･サービス･プロバイダーをサポート

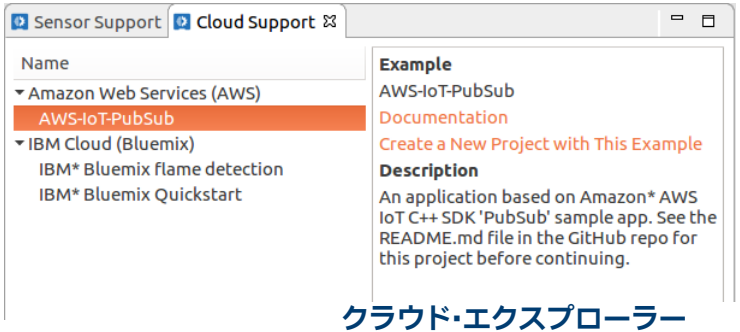
バージョン 2019 の IoT アプリケーション開発向け新機能

- **Yocto Project* 向けの Eclipse* プラグイン** - Yocto Project* 互換ターゲットのアプリケーション・プロジェクトおよびプラットフォーム・プロジェクトを作成またはインポート
- クラウド・サービス・プロバイダー (CSP) Google Cloud* IoT Core のサポートおよび CSP のコネクターを含むサンプル・プロジェクトの提供
- **MRAA IO コミュニケーション・レイヤー** - GPIO (汎用入出力) ドライバーを使用してオンボード LED を制御できる新しい API
  - 新しいプラットフォーム: IEI Tank、Ultra96*、Rock960、Hikey 960、Dragonboard* 820c
  - chardev GPIO デバイスと複数のボードでの実装をサポート - ほかのフレームワークとのシンプルな統合のために文字列ベースの I/O 初期化を追加
- **UPM センサーおよびアクチュエーター・ライブラリー** - JSON ライブラリー・ファイルのヘッダーとクラス名の doxygen タグの改良によりセンサー・エクスプローラーとの統合を促進
  - Java* パッケージベースの Cordova バインディングを追加
  - nmea_gps ライブラリーの全面的な変更により、データ取得と解析を細かく制御
  - 新しいセンサー: kx122、kxtj3、bh1792、bh1749

デバイスとクラウドを接続する統合開発エクスペリエンス

統合例





クラウド・エクスプローラー

- 分散したデバイスとクラウドを接続してデータを管理/解析
- Microsoft* Azure*、IBM* QuickStart、AWS* IOT Cloud、Google Cloud*
- 統合サンプル、クラウド・テクノロジー・エクスプローラー
- Arduino* Create (例、パッケージ・マネージャー、ライブラリー) をインテル® System Studio にインポート
- オペレーティング・システムと依存性を備えたコンテナーをプリインストールおよび動作検証済み

IBM BlueMix　Microsoft Azure　amazon web services™　Google Cloud Platform

ライブラリーを利用してセンサー機能を追加

400 を超える特殊センサードライバー、数千の基本プロトタイプと工業用センサーをサポート (例: LED、スイッチ)

- センサー・ライブラリー: ハードウェア抽象化レイヤー (mraa) とセンサー・フレームワーク (UPM)
- センサーやアクチュエーターを利用する開発作業を簡素化
- インテル® IoT 開発キットおよびインテル® プラットフォーム間の移植性
- 工業用センサーと接続プロトコルを含む 400 を超えるセンサーをサポート
- すべての API にサンプルコードとドキュメントがある
- 業界標準を補完

プロトタイプ

DFROBOT DRIVE THE FUTURE
seeed Grow the Difference
adafruit
sparkfun ELECTRONICS

産業

BOSCH
OMEGA®
OMET
AEOTEC BY AEON LABS
Honeywell

… ほか

センサー・フレームワークを使用する理由

例: 典型的な I2C センサーからデータを読み取る

センサー・フレームワークを使用しない場合

pinmux の設定 (約 200 行のコード)

センサーとのやり取り

センサー・フレームワークを使用する場合

(約 5 行のコード)

- MRAA が pinmux 処理とメモリー割り当てを行う
- UPM がデフォルトの設定を提供し、センサーを有効にする
- mraa 対応プラットフォームへ簡単にコードを移植可能

インテル® System Studio サンプル & チュートリアル

https://github.com/intel-system-studio/samples (英語)

https://software.intel.com/en-us/system-studio/documentation/code-samples (英語)

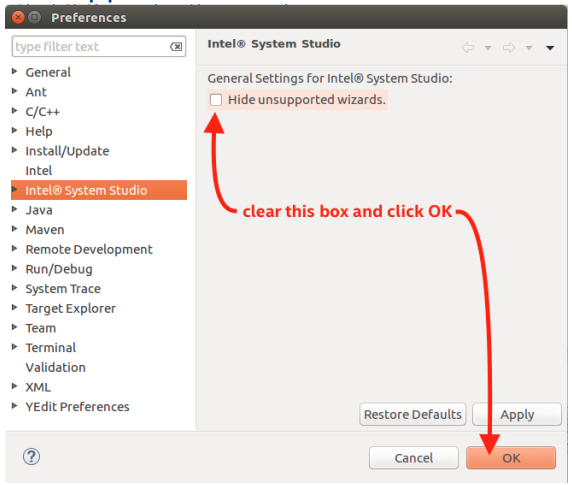
サンプルプログラム

https://github.com/intel-system-studio/samples (英語)

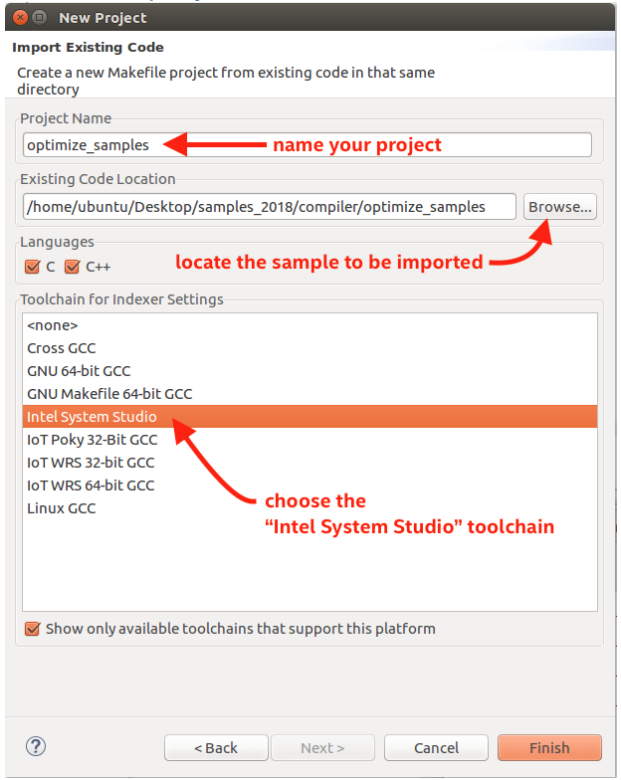
製品	サンプル
インテル® System Studio	Using Intel® System Studio Components for Matrix Multiplication Computation
インテル® C++ コンパイラー for Linux*	Hello World
	Averagingfilter_SDLT
	IPO sample
	Intrinsic
	Compiler Optimization Levels
	PGO optimization
	Threading Your Application - Tachyon
	Auto-Vectorization Sample
インテル® Enhanced GDB	GDB* Reverse Debugging Sample
インテル® IPP for Linux*	Tutorial: Blurring and Rotation Sample
インテル® System Debugger	Tutorial: Basic steps of debugging an application with the Intel® System Debugger
インテル® MKL for Linux*	Tutorial: Matrix Multiplication (C Language)
インテル® TBB for Linux*	Intel® TBB - example usages
インテル® VTune™ Amplifier for Linux*	Matrix Multiplication Sample
	Intel® Threading Building Blocks Tachyon Sample for Linux* targets
	Intel® Threading Building Blocks Tachyon Sample for Android* targets
	JIT Profiling Sample

サンプルプログラムのビルド (Eclipse*) (1/2)

“unsupported wizards” を有効にする



“makefile project” としてインポート



サンプルプログラムのビルド (Eclipse*) (2/2)

“toolchain build” 詳細の変更

Properties for optimize-samples-tony

Tool Chain Editor

Configuration: Default [Active] Manage Configurations...

Display compatible toolchains only

Current toolchain: Intel System Studio

Current builder: Gnu Make Builder

Gnu Make Builder

GNU Makefile 64-bit internal builder

Used tools

Intel Archiver
Intel C++ Compiler
Intel C Compiler
Intel C++ Linker
Intel C Linker

Properties for optimize-samples-tony

C/C++ Build

Builder Settings | Behavior | Refresh Policy

Builder

Builder type: External builder

Use default build command

Build command: make

Makefile generation

Generate Makefiles automatically

Expand Env. Variable Refs in Makefiles

Build location

Build directory: ${workspace_loc:/optimize_samples}/

Workspace... File system... Variables...

Restore Defaults Apply

Cancel OK

Properties for optimize-samples-tony

C/C++ Build

Configuration: Default [Active] Manage Configurations...

Builder Settings | Behavior | Refresh Policy

Build settings

Stop on first build error

Enable parallel build

Use optimal jobs (2)

Use parallel jobs: 2

Use unlimited jobs

Workbench Build Behavior

Workbench build type:

Make build target:

Build on resource save (Auto build) all Variables...

Note: See Workbench automatic build preference

Build (Incremental build) Variables...

Clean clean Variables...

Restore Defaults Apply

change from “all” to blank

Cancel OK

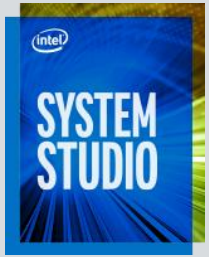
チュートリアル

https://software.intel.com/en-us/system-studio/documentation/code-samples

製品	チュートリアル
Intel® VTune™ Amplifier	Enable Performance Collection on an Embedded Linux* System
	Find Hotspots with Intel VTune Amplifier for Linux
	Analyze Energy Use on a Windows* Platform
	Analyze Locks and Waits with Intel VTune Amplifier
	Identify Hardware-Related Performance Issues with Intel VTune Amplifier
	Identify False Sharing with Intel VTune Amplifier for Linux
	Analyze Disk I/O Waits with Intel VTune Amplifier for Linux
	Overview of Systems Usage Models in Intel VTune Amplifier
Intel® Inspector	**Analyze Memory Errors with the Intel® Inspector**
Intel® MKL	Use Intel Math Kernel Library for Matrix Multiplication (C Language)
Intel® IPP	Blur and Rotate Images with Intel Integrated Performance Primitives
Intel® System Debugger	Get Deep Insight with Intel® Processor Trace
Intel® C++ Compiler	Use the Intel C++ Compiler for Embedded Linux Systems
Intel® MKL and Intel® IPP	Compare Signal Processing

開発者向けリソースの包括的なポートフォリオ

システム/IoT ツール



開発者向けキット (英語)

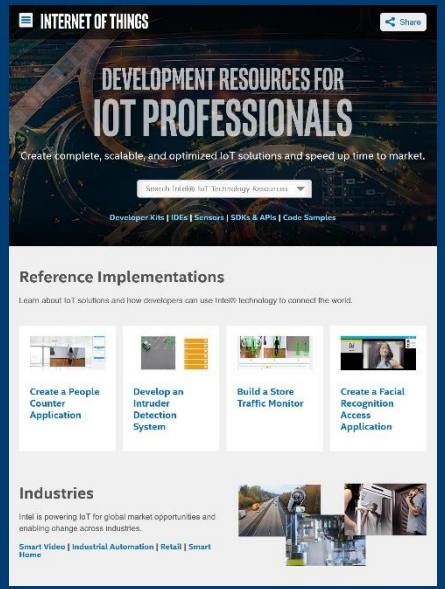
独自のツール

インテル® OpenVINO™ ツールキット (英語)
コンピューター・ビジョンを革新、ディープラーニングを統合

インテル® Media SDK
高速で高品質なビデオ・トランスコードを実現

インテル® Context Sensing SDK (英語)
コンテキストを意識した機能とサービスを開発

インテル® デベロッパー・ゾーン



software.intel.com/iot (英語)

プライオリティー・サポートの利用

利点

- **パフォーマンスと生産性を向上** - インテル® プロセッサーでコードのパフォーマンスを最大限に引き出す方法、パフォーマンス・ボトルネックや開発の課題を解消する方法などを問い合わせ可能
- インテルのエンジニアに**直接問い合わせることができ**、機密の問い合わせやコードサンプルを送信可能
- 技術的な質問やその他の製品ニーズに対する**ヘルプ**
- 製品の新しいアップデートおよび以前のバージョンへの**無料アクセス**
- サポートサービス更新時の大幅な割引

SlashData の年間アワードを受賞
Developer Choice Awards 2018 (英語)
(世界中の 2 万人を超える開発者による投票)

すべての開発者向けの関連情報

- **エキスパート・コミュニティー・サポート・フォーラム (英語)**
- 過去数十年のハイパフォーマンス・コード作成の経験を基に構築された**ドキュメント・ライブラリー**

インテル® System Studio には、エキスパート・コミュニティー・フォーラムによるサポートを利用できる無償のコミュニティー・ライセンスが用意されています。有償ライセンスにはプライオリティー・サポートが含まれます。(一部のインテルのお客様もプライオリティー・サポートを利用できます。)

さあ、始めよう

インテル® System Studio のダウンロード

- 無償のコミュニティー・ライセンス (英語)
- 学生 (英語)/学術研究者 (英語) 向け無料/割引バージョン

開発者向けリソース

- 製品サイト
- 入門ガイド/ドキュメント (英語)
- サンプルコード/トレーニング (英語)
- Tech.Decoded ウェビナー、HOW TO ビデオと記事 (英語)
- エキスパート・コミュニティー・サポート・フォーラム (英語)

有償ライセンスにはプライオリティー・サポートが含まれます

Q & A

補足資料

システムワイドのクローズドシャーシのデバッグ

低コストの USB 接続による JTAG ベースのシステムデバッグとシステムトレース

- 開発者がエンジニアリング･サンプルではなく、実際のプロダクション･ハードウェア上で作業する時間を排除
- JTAG 機器/プローブにかかるコストを節約 - 高価な JTAG プローブの代わりに標準 USB 接続でデバッグ
- JTAG デバッグ･ステーションを交代で使用する必要なし
- 柔軟な設計によりアクセス可能なハードウェア JTAG ポートの要件を緩和

USB ケーブル

OS ブートのデバッグ & トレース

インテル® SVT Closed Chassis Adapter[1]

CPU リセットからのデバッグ & トレース

[1] SVT = Silicon View Technology – 詳細: https://designintools.intel.com/product_p/itpxdpsvt.htm (英語)

インテルの OpenCL* ツールで開発を行う理由

パフォーマンス

向上

- パフォーマンス/ワット
- コードのカスタマイズ
- スケーラビリティー

スピードと効率

短縮/削減

- 開発期間
- 市場投入までの時間
- 総所有コスト

アクセラレーター・ハードウェアの活用

CPU

GPU

独自のアプリケーションを開発

システム、IoT/組込みアプリケーション

ビジュアル計算アプリケーション

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。

インテルの OpenCL* ツールで開発を行う理由

さまざまなインテル® アーキテクチャー・ベースのデバイス (CPU および GPU - 低レベルでアクセス) で **'独自の'** アルゴリズムを開発。
インテル® System Studio のツールとライブラリーを使用して開発を高速化。
ほかのインテル® ソフトウェア開発ツール (インテル® Media SDK、インテル® OpenVINO™ ツールキット、その他) と簡単に統合。

競争優位性の確立、より簡単なヘテロジニアス・アプローチ

- OpenCL* フレームワークを使用した低レベル開発により競争力に優れた IP ベースのアプリケーションを開発。
- 共通のフロントエンド (OpenCL* ツール) を使用してインテル® アーキテクチャー・ベースのデバイス (CPU および GPU) 向けのコードを作成。

生産性と効率の向上

- 複数のハードウェア・タイプでカスタム開発。パフォーマンス・ポータブルではないが、ハードウェア固有のスタックよりも新しいハードウェアの開発方法の習得にかかる時間が短い。
- カスタム・アルゴリズムによりインテル® Media SDK およびインテル® OpenVINO™ ツールキットで記述されたパイプラインを OpenCL* を使用して拡張。
- インテル® System Studio に含まれるアナライザー (インテル® VTune™ Amplifier など) によりシステムのパフォーマンスと電力効率を最適化。
- インテル® System Studio に含まれるライブラリー (インテル® MKL、インテル® IPP、インテル® TBB など) を活用してカーネル開発を合理化、最適化を向上。

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。

インテル® System Studio のエディションと構成

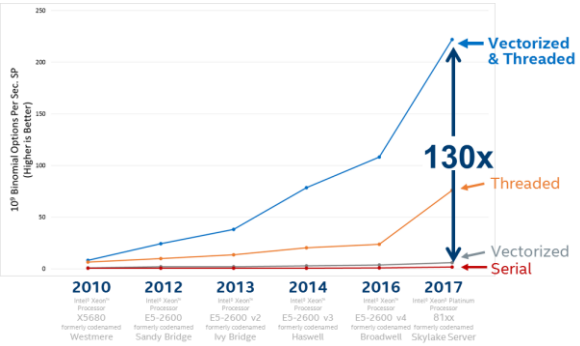
	無料ダウンロード	インテル® System Studio Composer Edition	インテル® System Studio Professional Edition	インテル® System Studio Ultimate Edition
ビルド	**インテル® C++ コンパイラー**	√	√	√
	インテル® マス・カーネル・ライブラリー (インテル® MKL)	√	√	√
	インテル® データ・アナリティクス・アクセラレーション・ライブラリー (インテル® DAAL)	√	√	√
	インテル® インテグレーテッド・パフォーマンス・プリミティブ (インテル® IPP)	√	√	√
	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)	√	√	√
	OpenCL* ツール	√	√	√
	IoT 接続ツール (UPM/MRAA/クラウドコネクター)	√	√	√
解析	**インテル® VTune™ Amplifier** - パフォーマンス・プロファイラー		√	√
	電力解析		√	√
	インテル® Advisor		√	√
	インテル® Inspector – メモリー/スレッドのデバッガー		√	√
	インテル® Graphics Performance Analyzers		√	√
デバッグ	**インテル® System Debugger**			√
	インテル® Debug Extensions for WinDbg			√
	プライオリティー・サポート - インテルのエンジニアに直接問い合わせ	√	√	√[1]
サポート	**ホスト・オペレーティング・システム**	Linux*、Windows*、macOS*		
	ターゲット OS	**ターゲット OS システム:** Linux*、Linux* + Android* (コンボ)、Windows*、FreeBSD* **OS サポート:** Linux*、組込み Linux*、Windows*、Wind River* Linux*、Yocto Project*、Android*、Wind River* VXWorks*		
	IDE	Eclipse* Standard、Microsoft* Visual Studio*、Wind River* Workbench		
	言語	C/C++		
	サポートされるプロセッサー[2]	Intel Atom® プロセッサー、インテル® Core™ プロセッサー、インテル® Xeon® プロセッサー		

FreeBSD* バージョンとその他の構成: フローティング、無償、学生/アカデミック向けの割引バージョン、および Wind River* VxWorks* サポートについては、www.xlsoft.com/jp/products/download/intelj.html を参照してください。
[1] Ultimate Edition の無償の 90 日間の更新可能な商用ライセンス (英語) では、コミュニティー・フォーラムによるサポートが提供されます。プライオリティー・サポートは、有償ライセンスにのみ含まれます。
[2] サポートされるプロセッサーの詳細は、リリースノートを参照してください。

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。

ベクトル化およびスレッド化するか、パフォーマンスを諦めるか

スレッド化 + ベクトル化はどちらか一方よりもはるかに高速



2010 年 ~ 2017 年のベンチマークのシステム構成

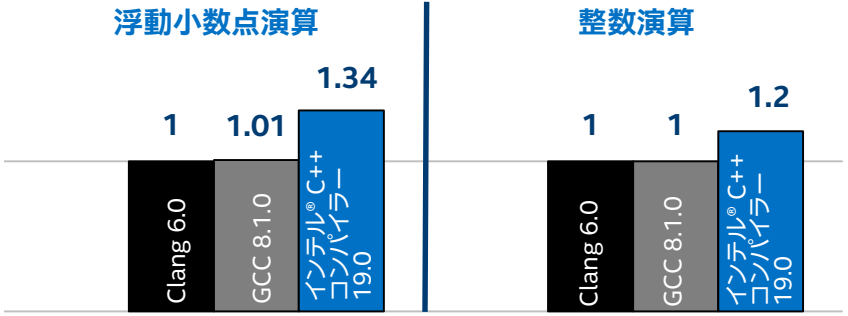
プラットフォーム、ハードウェア、ソフトウェア

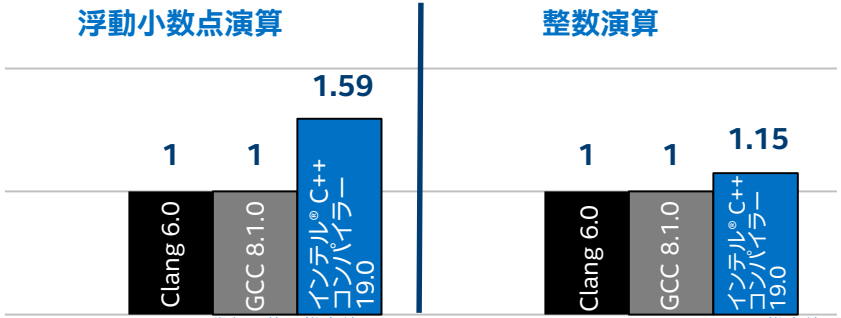
	プラットフォーム	スケーリングされていないコアクロックの周波数	コア/ソケット	ソケット数	L1データキャッシュ	L2キャッシュ	L3キャッシュ	メモリー	メモリー周波数	メモリーアクセス	H/W プリフェッチ有効	HT有効	ターボ有効	Cステート	OS	カーネル	コンパイラー
WSM†	インテル® Xeon™ プロセッサー X5680	3.33GHz	6	2	32K	256K	12MB	48MB	1333MHz	NUMA	Y	Y	Y	無効	Fedora* 20	3.11.10-301.fc20	icc 17.0.2
SNB†	インテル® Xeon® プロセッサー E5 2690	2.90GHz	8	2	32K	256K	20MB	64GB	1600MHz	NUMA	Y	Y	Y	無効	Fedora* 20	3.11.10-301.fc20	icc 17.0.2
IVB†	インテル® Xeon® プロセッサー E5 2697 v2	2.70GHz	12	2	32K	256K	30MB	64GB	1867MHz	NUMA	Y	Y	Y	無効	RHEL 7.1	3.10.0-229.el7.x86_64	icc 17.0.2
HSW†	インテル® Xeon® プロセッサー E5 2699 v3	2.30GHz	18	2	32K	256K	46MB	128GB	2133MHz	NUMA	Y	Y	Y	無効	Fedora* 20	3.15.10-200.fc20.x86_64	icc 17.0.2
BDW†	インテル® Xeon® プロセッサー E5 2697 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 17.0.2
BDW†	インテル® Xeon® プロセッサー E5 2699 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 17.0.2
SKX†	インテル® Xeon® Platinum 81xx プロセッサー	2.50GHz	28	2	32K	1024K	40MB	192GB	2666MHz	NUMA	Y	Y	Y	無効	CentOS* 7.3	3.10.0-514.10.2.el7.x86_64	icc 17.0.2

ベンチマーク: 2 項オプション価格モデルhttps://software.intel.com/en-us/articles/binomial-options-pricing-model-code-for-intel-xeon-phi-coprocessor (英語)。
パフォーマンス結果は 2017 年 8 月時点のテスト結果に基づいたものであり、公開されている利用可能なすべてのセキュリティー・アップデートが適用されていない可能性があります。詳細については、構成の開示を参照してください。絶対的なセキュリティーを提供できる製品はありません。性能やベンチマーク結果について、さらに詳しい情報をお知りになりたい場合は、http://www.intel.com/benchmarks/ (英語) を参照してください。2017 年 8 月時点のインテルによるテスト。システム構成は上記を参照。インテル® コンパイラーでは、インテル® マイクロプロセッサーに限定されない最適化に関して、他社製マイクロプロセッサー用に同等の最適化を行えないことがあります。これには、インテル® ストリーミング SIMD 拡張命令 2、インテル® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサーに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサー依存の最適化は、インテル® マイクロプロセッサーでの使用を前提としています。インテル® マイクロアーキテクチャーに限定されない最適化のなかにも、インテル® マイクロプロセッサー用のものがあります。この注意事項で言及した命令セットの詳細については、該当する製品のユーザー・リファレンス・ガイドを参照してください。注意事項の改訂 #20110804

†開発コード名

インテル® Xeon® プロセッサーにおいて SPEC* ベンチマークで優れたシステム・パフォーマンスを達成 - インテル® C++ コンパイラー





パフォーマンス結果は 2018 年 8 月 26 日時点のテスト結果に基づいたものであり、公開されている利用可能なすべてのセキュリティー・アップデートが適用されていない可能性があります。詳細については、構成の開示を参照してください。絶対的なセキュリティーを提供できる製品はありません。性能に関するテストに使用されるソフトウェアとワークロードは、性能がインテル® マイクロプロセッサー用に最適化されていることがあります。SYSmark* や MobileMark* などの性能テストは、特定のコンピューター・システム、コンポーネント、ソフトウェア、操作、機能に基づいて行ったものです。結果はこれらの要因によって異なります。製品の購入を検討される場合は、他の製品と組み合わせた場合の本製品の性能など、ほかの情報や性能テストも参考にして、パフォーマンスを総合的に評価することをお勧めします。詳細については、パフォーマンス・ベンチマーク・テストの開示 (英語) を参照してください。

2018 年 8 月 26 日時点のインテルによるテスト。**システム構成:** ハードウェア: インテル® Xeon® Platinum 8180 プロセッサー @ 2.50GHz、384GB RAM、ハイパースレッディング有効。ソフトウェア: インテル® コンパイラー 19.0、GCC 8.1.0、PGI* 18.5、Clang/LLVM 6.0。Linux* OS: Red Hat* Enterprise Linux* Server 7.4 (Maipo)、3.10.0-693.el7.x86_64。SPEC* ベンチマーク (www.spec.org (英語))。SPECint* ベンチマーク測定時の CXX テストには SmartHeap* 10 を使用。SPECint*_rate_base_2017 コンパイラー・オプション: C++ テストには SmartHeap* 10 を使用。インテル® C/C++ コンパイラー 19.0: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-mem-layout-trans=3。GCC 8.1.0: -march=znver1 -mfpmath=sse -Ofast -funroll-loops -flto。Clang 6.0: -march=core-avx2 -mfpmath=sse -Ofast -funroll-loops –flto。SPECfp*_rate_base_2017 コンパイラー・オプション: インテル® C/C++ コンパイラー 19.0: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-prefetch -ffinite-math-only -qopt-mem-layout-trans=3。GCC 8.1.0: -march=skylake-avx512 -mfpmath=sse -Ofast -fno-associative-math -funroll-loops -flto。Clang 6.0: -march=znver1 -mfpmath=sse -Ofast -funroll-loops –flto。SPECint*_speed_base_2017 コンパイラー・オプション: C++ テストには SmartHeap* 10 を使用。インテル® C/C++ コンパイラー 19.0: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-mem-layout-trans=3 -qopenmp。GCC 8.1.0: -march=znver1 -mfpmath=sse -Ofast -funroll-loops -flto -fopenmp。Clang 6.0: -march=core-avx2 -mfpmath=sse -Ofast -funroll-loops -flto -fopenmp=libomp SPECfp*_speed_base_2017 コンパイラー・オプション: インテル® C/C++ コンパイラー 19.0: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-prefetch -ffinite-math-only -qopenmp。GCC 8.1.0: -march=skylake-avx512 -mfpmath=sse -Ofast -fno-associative-math -funroll-loops -flto -fopenmp。Clang 6.0: -march=skylake-avx512 -mfpmath=sse -Ofast -funroll-loops -flto -fopenmp=libomp。

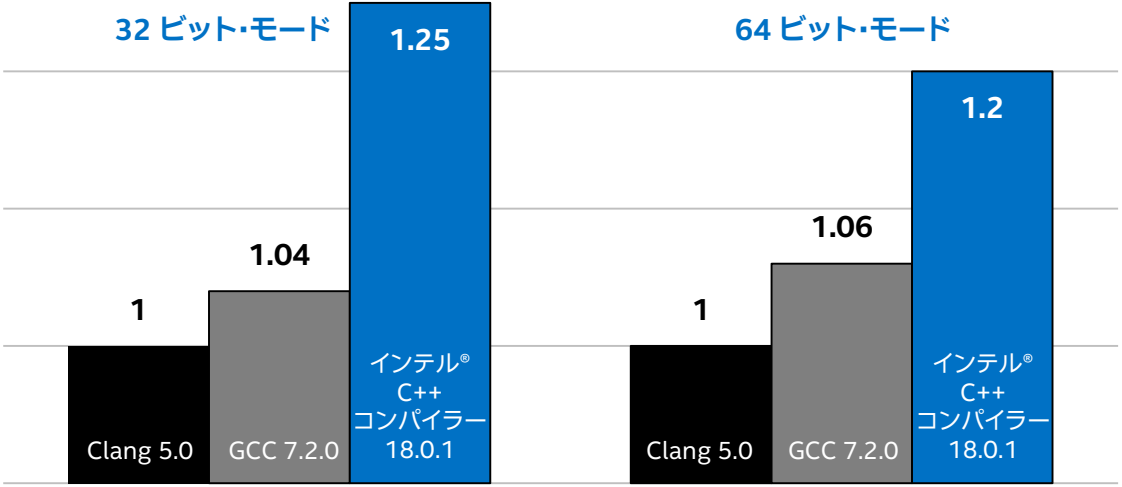
2018 年 8 月 26 日時点のインテルによるテスト。**システム構成:** ハードウェア: インテル® Xeon® Platinum 8180 プロセッサー @ 2.50GHz、384GB RAM、ハイパースレッディング有効。ソフトウェア: インテル® コンパイラー 19.0、GCC 8.1.0、PGI* 18.5、Clang/LLVM 6.0。Linux* OS: Red Hat* Enterprise Linux* Server 7.4 (Maipo)、3.10.0-693.el7.x86_64。SPEC* ベンチマーク (www.spec.org (英語))。SPECint* ベンチマーク測定時の CXX テストには SmartHeap* 10 を使用。SPECint*_rate_base_2017 コンパイラー・オプション: C++ テストには SmartHeap* 10 を使用。インテル® C/C++ コンパイラー 19.0: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-mem-layout-trans=3。GCC 8.1.0: -march=znver1 -mfpmath=sse -Ofast -funroll-loops -flto。Clang 6.0: -march=core-avx2 -mfpmath=sse -Ofast -funroll-loops –flto。SPECfp*_rate_base_2017 コンパイラー・オプション: インテル® C/C++ コンパイラー 19.0: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-prefetch -ffinite-math-only -qopt-mem-layout-trans=3。GCC 8.1.0: -march=skylake-avx512 -mfpmath=sse -Ofast -fno-associative-math -funroll-loops -flto。Clang 6.0: -march=znver1 -mfpmath=sse -Ofast -funroll-loops –flto。SPECint*_speed_base_2017 コンパイラー・オプション: C++ テストには SmartHeap* 10 を使用。インテル® C/C++ コンパイラー 19.0: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-mem-layout-trans=3 -qopenmp。GCC 8.1.0: -march=znver1 -mfpmath=sse -Ofast -funroll-loops -flto -fopenmp。Clang 6.0: -march=core-avx2 -mfpmath=sse -Ofast -funroll-loops -flto -fopenmp=libomp。SPECfp*_speed_base_2017 コンパイラー・オプション: インテル® C/C++ コンパイラー 19.0: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-prefetch -ffinite-math-only -qopenmp。GCC 8.1.0: -march=skylake-avx512 -mfpmath=sse -Ofast -fno-associative-math -funroll-loops -flto -fopenmp。Clang 6.0: -march=skylake-avx512 -mfpmath=sse -Ofast -funroll-loops -flto -fopenmp=libomp。

XLSOFT

インテル® コンパイラーでは、インテル® マイクロプロセッサーに限定されない最適化に関して、他社製マイクロプロセッサー用に同等の最適化を行えないことがあります。これには、インテル® ストリーミング SIMD 拡張命令 2、インテル® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサーに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサー依存の最適化は、インテル® マイクロプロセッサーでの使用を前提としています。インテル® マイクロアーキテクチャーに限定されない最適化のなかにも、インテル® マイクロプロセッサー用のものがあります。この注意事項で言及した命令セットの詳細については、該当する製品のユーザー・リファレンス・ガイドを参照してください。注意事項の改訂 #20110804

Intel Atom® プロセッサーにおいて組込みアプリケーションのパフォーマンスを大幅に向上 - インテル® C++ コンパイラー

Intel Atom® プロセッサー C3958 上での Coremark*-Pro ベンチマーク



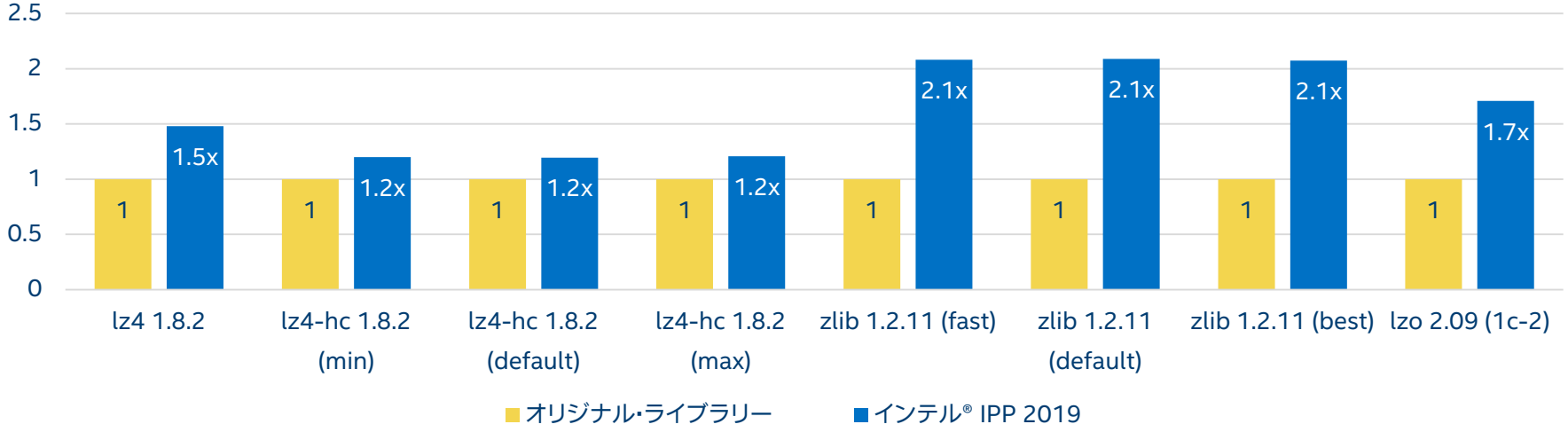
ベンチマーク結果は、「Spectre」および「Meltdown」と呼ばれる脆弱性への対処を目的とした最新のソフトウェア・パッチおよびファームウェア・アップデートの適用前に取得されたものです。パッチやアップデートを適用したデバイスやシステムでは同様の結果が得られないことがあります。性能やベンチマーク結果について、さらに詳しい情報をお知りになりたい場合は、www.intel.com/benchmarks (英語) を参照してください。
システム構成: Intel Atom® プロセッサー C3958 @ 2.00GHz、64GB RAM、インテル® ハイパースレッディング・テクノロジー無効。ソフトウェア: インテル® C++ コンパイラー 18.0.1、GCC 7.2.0、Clang/LLVVM 5.0。
Linux* OS: Fedora* 25 (Twenty Five)、カーネル 4.8.6-300.fc25.x86_64。Coremark*-Pro ベンチマーク (www.eembc.org (英語))。
コンパイラー・オプション: インテル® C++ コンパイラー 18.0.1: -O3 -no-prec-div -ipo -xATOM_SSE4.2 -mtune=goldmont.。GCC 7.2.0: -Ofast -mfpmath=sse -flto -march=native -funroll-loops -ffat-lto-objects。
Clang/LLVM 5.0: -Ofast -mfpmath=sse -flto -march=native -funroll-loops。GCC および Clang/LLVM の 32 ビット・モードでは、-m32 オプションを追加。ベンチマークの出典: インテル コーポレーション。

インテル® コンパイラーでは、インテル® マイクロプロセッサーに限定されない最適化に関して、他社製マイクロプロセッサー用に同等の最適化を行えないことがあります。これには、インテル® ストリーミング SIMD 拡張命令 2、インテル® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサーに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサー依存の最適化は、インテル® マイクロプロセッサーでの使用を前提としています。インテル® マイクロアーキテクチャーに限定されない最適化のなかにも、インテル® マイクロプロセッサー用のものがあります。この注意事項で言及した命令セットの詳細については、該当する製品のユーザー・リファレンス・ガイドを参照してください。注意事項の改訂 #20110804

データ展開のパフォーマンスを向上

インテル® IPP

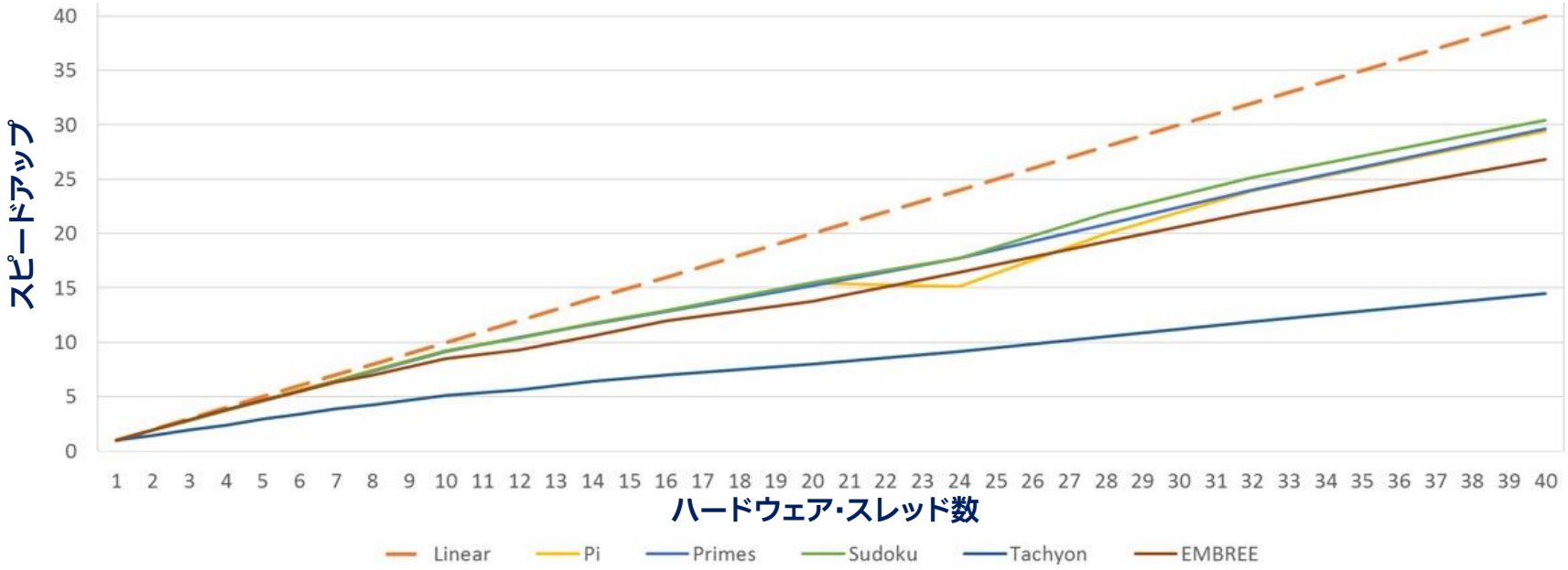
インテル® IPP 2019 と LZ4、Zlib、LZO ライブラリーの データ展開パフォーマンスの比較



パフォーマンス結果は 2018 年 8 月 15 日時点のテスト結果に基づいたものであり、公開されている利用可能なすべてのセキュリティー・アップデートが適用されていない可能性があります。詳細については、構成の開示を参照してください。絶対的なセキュリティーを提供できる製品はありません。
性能に関するテストに使用されるソフトウェアとワークロードは、性能がインテル® マイクロプロセッサー用に最適化されていることがあります。SYSmark* や MobileMark* などの性能テストは、特定のコンピューター・システム、コンポーネント、ソフトウェア、操作、機能に基づいて行ったものです。結果はこれらの要因によって異なります。製品の購入を検討される場合は、他の製品と組み合わせた場合の本製品の性能など、ほかの情報や性能テストも参考にして、パフォーマンスを総合的に評価することをお勧めします。詳細については、パフォーマンス・ベンチマーク・テストの開示 (英語) を参照してください。
2018 年 8 月 15 日時点のインテルによるテスト。システム構成: ハードウェア: インテル® Core™ i5-7600 プロセッサー @ 3.50GHz、4 コア、ハイパースレッディング無効。キャッシュ: L1=32KB、L2=256KB、L3=6MB。メモリー: 64GB。Linux* OS: Red Hat* Enterprise Linux* Server 7.2。

XLSOFT
インテル® コンパイラーでは、インテル® マイクロプロセッサーに限定されない最適化に関して、他社製マイクロプロセッサー用に同等の最適化を行えないことがあります。これには、インテル® ストリーミング SIMD 拡張命令 2、インテル® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサーに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサー依存の最適化は、インテル® マイクロプロセッサーでの使用を前提としています。インテル® マイクロアーキテクチャーに限定されない最適化のなかにも、インテル® マイクロプロセッサー用のものがあります。この注意事項で言及した命令セットの詳細については、該当する製品のユーザー・リファレンス・ガイドを参照してください。注意事項の改訂 #20110804

インテル® TBB 2019 の優れたパフォーマンス・スケーラビリティー - インテル® Xeon® プロセッサー



パフォーマンス結果は 2018 年 7 月 31 日時点のテスト結果に基づいたものであり、公開されている利用可能なすべてのセキュリティー・アップデートが適用されていない可能性があります。詳細については、構成の開示を参照してください。絶対的なセキュリティーを提供できる製品はありません。性能やベンチマーク結果について、さらに詳しい情報をお知りになりたい場合は、http://www.intel.com/benchmarks/ (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、性能がインテル® マイクロプロセッサー用に最適化されていることがあります。SYSmark* や MobileMark* などの性能テストは、特定のコンピューター・システム、コンポーネント、ソフトウェア、操作、機能に基づいて行ったものです。結果はこれらの要因によって異なります。製品の購入を検討される場合は、他の製品と組み合わせた場合の本製品の性能など、ほかの情報や性能テストも参考にして、パフォーマンスを総合的に評価することをお勧めします。詳細については、www.intel.com/benchmarks (英語) を参照してください。2018 年 7 月 31 日時点のインテルによるテスト。システム構成: ソフトウェア: インテル® 64 対応インテル® C++ コンパイラー 18.0、インテル® TBB 2019。ハードウェア: 2x インテル® Xeon® Gold 6152 プロセッサー @ 2.10GHz、192GB RAM。Linux* OS: CentOS* 7.4 1708 (Core)、カーネル 3.10.0-693.e17.x86_64。注: sudoku、primes、tachyon はインテル® TBB に含まれているコードを使用。

インテル® コンパイラーでは、インテル® マイクロプロセッサーに限定されない最適化に関して、他社製マイクロプロセッサー用に同等の最適化を行えないことがあります。これには、インテル® ストリーミング SIMD 拡張命令 2、インテル® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサーに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサー依存の最適化は、インテル® マイクロプロセッサーでの使用を前提としています。インテル® マイクロアーキテクチャーに限定されない最適化のなかにも、インテル® マイクロプロセッサー用のものがあります。この注意事項で言及した命令セットの詳細については、該当する製品のユーザー・リファレンス・ガイドを参照してください。注意事項の改訂 #20110804

法務上の注意書きと最適化に関する注意事項

パフォーマンス結果は 2018 年 9 月時点のテスト結果に基づいたものであり、公開されている利用可能なすべてのセキュリティー・アップデートが適用されていない可能性があります。詳細については、構成の開示を参照してください。絶対的なセキュリティーを提供できる製品はありません。

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

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。

© 2018 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、Intel Atom、Celeron、Intel Core、Iris、Pentium、Xeon、Cilk、OpenVINO、VTune は、アメリカ合衆国および / またはその他の国における Intel Corporation またはその子会社の商標です。

* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

最適化に関する注意事項

インテル® コンパイラーでは、インテル® マイクロプロセッサーに限定されない最適化に関して、他社製マイクロプロセッサー用に同等の最適化を行えないことがあります。これには、インテル® ストリーミング SIMD 拡張命令 2、インテル® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサーに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサー依存の最適化は、インテル® マイクロプロセッサーでの使用を前提としています。インテル® マイクロアーキテクチャーに限定されない最適化のなかにも、インテル® マイクロプロセッサー用のものがあります。この注意事項で言及した命令セットの詳細については、該当する製品のユーザー・リファレンス・ガイドを参照してください。
注意事項の改訂 #20110804