



マルチスレッド・ライブラリーを使用した デジタル・メディア・アプリケーションの パフォーマンスの向上

Ying Song

テクニカル・コンサルティング・エンジニア
インテル® ソフトウェア開発製品



並列化のステップ – マルチコア・プログラミングのための 8 つの規則



1. 抽象化を使用してプログラミングする
2. スレッドではなく、タスクをプログラミングする
3. パラレルに考える
4. 並列処理を無効にするオプションを設計する
5. ロックのタイミングとバランスのとれたロックの使用方法を習得する
6. 並列処理を支援するツールやライブラリーを使用する
7. スケーラブルなメモリー・アロケータを使用する
8. ワークロードを増やしてスケーラビリティが向上するように設計する



インテル® ソフトウェア開発製品で実現する並列化

インテル® VTune™ パフォーマンス・アナライザー

- ソースコード中のパフォーマンスのボトルネックを識別し、マルチコア・パフォーマンスを最適化

インテル® コンパイラー

- インテル® プロセッサのパフォーマンスを最大限に引き出すアプリケーションを開発する最良の方法

インテル® パフォーマンス・ライブラリー

- 高度に最適化されたスレッドセーフなマルチメディア関数と HPC 数値演算関数

インテル® スレッディング・ビルディング・ブロック

- パフォーマンスとスケーラビリティを備えたマルチスレッド・アプリケーションの作成を単純化する C++ テンプレート・ベースのランタイム・ライブラリー

インテル® スレッド化解析ツール

- スレッド化エラーを検出し、マルチスレッド・アプリケーションを最適化して最高レベルの性能を実現

インテル® クラスターツール

- クラスタベースのアプリケーションの作成、解析、最適化および運用

<http://www.intel.co.jp/jp/software/products/>



マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上



このセッションの内容

- 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) の概要
- インテル® IPP とマルチスレッディング
- インテル® IPP バージョン 5.3 - **最新リリース**
- 要約



インテル® IPP の概要



製品コンポーネント

アプリケーションのソースコード

インテル® IPP 使用法コードサンプル

- サンプルのビデオ/オーディオ/音声コーデック
- 画像処理および JPEG
- 信号処理
- データ圧縮
- 暗号化
- .NET* および Java* の統合



アプリケーション
開発の高速化



↓ ↓ ↓ ↓ ↓ API 呼び出し

インテル® IPP ライブラリー C/C++ API

- ビデオ・コーディング
- オーディオ・コーディング
- 音声コーディング
- 音声認識
- データ圧縮
- 暗号化
- 行列演算
- 信号処理
- 画像処理
- JPEG および JPEG 2000
- コンピューター・ビジョン
- 画像カラー変換
- スtring処理
- ベクトル演算
- リアルスティック・レンダリング



クロスプラット
フォームの互換性
コードの再利用



↓ ↓ ↓ ↓ ↓ スタティック/ダイナミック・リンク

各種プロセッサ用に最適化された インテル® IPP バイナリー

- インテル® Core™ プロセッサ・ファミリー
- インテル® Pentium® D プロセッサ
- インテル® Pentium® M プロセッサ
- インテル® Xeon® プロセッサ
- インテル® Itanium® プロセッサ
- Intel XScale® テクノロジー・ベースのプロセッサ



優れた
パフォーマンス

マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上





インテル® IPP の広範囲な関数とサンプル (I)

ドメイン	関数	サンプル
1. 画像処理	<ul style="list-style-type: none">* サイズ変更/回転などのジオメトリ変換* 画像の線形および非線形フィルタリング操作: エッジ検出、ブラー、ノイズ除去などのフィルター効果用* 2D 線形変換 FFT、DFT、DCT* 画像統計および解析	* タイル画像処理、2D ウェーブレット変換、C++ 画像処理クラス、画像処理関数デモ
2. カラー変換	<ul style="list-style-type: none">* 画像/ビデオのカラー空間フォーマット変換: RGB、HSV、YUV、YCbCr* アップ/ダウンサンプリング* 輝度およびコントラストの調整	
3. JPEG コーディング	<ul style="list-style-type: none">* 高レベル JPEG/JPEG 2000 圧縮および解凍関数* JPEG/JPEG 2000 サポート関数: DCT、ウェーブレット変換、カラー変換、ダウンサンプリング	* インテル® JPEG ライブラリー (IJL) との統合、独立 JPEG グループ (IJG) ライブラリーとの統合、JPEG 2000 エンコーダー、デコーダー、JPEG ビューアー
4. ビデオ・ コーディング	<ul style="list-style-type: none">* VC-1、H.264、AVS、MPEG-2、MPEG-4、H.261、H.263 および DV コーデックサポート関数	* シンプル・メディア・プレーヤー、ビデオ・エンコーダー、H.264、DV デコーディング、ビデオ・トランスコーディング、反射デモ、仮想マルチチャンネル・オーディオ・プレーヤー
5. コンピューター・ ビジョン	<ul style="list-style-type: none">* 背景差分、特徴検出 (コーナー検出、エッジ検出)、距離変換、画像勾配、フラッドフィル、動き解析およびオブジェクト・トラッキング、角錐、パターン認識、カメラ・キャリブレーション	* 顔検出
6. リアリスティック・ レンダリング	<ul style="list-style-type: none">* アクセラレーション構造、レイシーン交差およびレイトレーシング* サーフェス・プロパティ、シェーダーサポート、トーン・マッピング	* レイトレーシング

インテル® IPP は広範囲なアプリケーションに最適

マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上





インテル® IPP の広範囲な関数とサンプル (II)

ドメイン	関数	サンプル
7. 信号処理	<ul style="list-style-type: none"> * 変換: DCT、DFT、MDCT、ウェーブレット (Haar とユーザー定義フィルターバンク)、ヒルベルト * 畳み込み、相互相関、自己相関、共役 * フィルタリング: IIR/FIR/メディアン・フィルタリング、シングル/マルチレート FIR LMS フィルター * その他: 窓、Jaehne/Tone/Triangle 信号生成、しきい値 	* 信号処理関数デモ
8. オーディオ・コーディング	* MP3、AAC、HE-AAC、AC3	* オーディオ・コーデック・コンソール・アプリケーション
9. 音声コーディング	* 適応/固定コードブック関数、自己相関、畳み込み、Levinson-Durbin 再帰、線形予測解析および量子化、エコー・キャンセラー、圧伸	* G.168、G.167、G.711、G.722、G.722.1、G.722.2、AMRWB、拡張 AMRWB (AMRWB+)、G.723.1、G.726、G.728、G.729、RT-Audio、GSM AMR、GSM FR
10. 音声認識	* 特徴処理、モデル評価/推定/適応、ベクトル量子化、多相再サンプリング、アドバンスト・オーロラ、Ephraim-Malah ノイズ・サプレッサー、AEC、音声検出	* オーロラ、アドバンスト・オーロラ、オーディオ処理、混合ガウス分布、音声処理
11. データ圧縮	<ul style="list-style-type: none"> * エントロピー・コーディング圧縮: ホフマン、VLC * 辞書圧縮: LZSS、LZ77 * Burrows-Wheeler 変換 (BWT)、Move-To-Front (MTF)、RLE (Run-Length Encoding)、Generalized Interval Transformation (GIT) * zlib および bzip2 の互換機能サポート 	* zlib、bzip2、gzip 互換 / 一般的なデータ圧縮例
12. 暗号化	* 多倍長算術 / ラインダール、DES、TDES、SHA1、MD5、RSA、DSA、モンゴメリー、素数生成および疑似乱数生成 (PRNG) 関数	* OpenSSL におけるインテル® IPP 暗号化の使用
13. スtring処理	* 比較、挿入、大文字小文字の変換、トリム、検索、正規表現、ハッシュ	* “ippgrep” – 正規表現マッチング

マルチスレッド

インテル® IPP は広範囲なアプリケーションに最適





インテル® IPP の広範囲な関数とサンプル (III)

ドメイン	関数	サンプル
14. ベクトル演算	* 論理、シフト、変換、累乗、根、指数、対数、三角、双曲線、Erf、Erfc	
15. 行列演算	* 加算、乗算、分解、固有値、積和、転置	
その他の共通関数	* プロセッサの種類、スレッド数制御、メモリー割り当て	* リンク/異なる言語のサポート * CPUID サンプル

インテル® IPP は非常に広範囲のアプリケーションに最適

- ビデオ・ブロードキャスト、ビデオ/音声会議
- コンシューマー・マルチメディア
- 医療用画像、文書用画像
- コンピューター・ビジョン/オブジェクト・トラッキング/マシンラーニング
- データベースおよびエンタープライズ・データ・マネージメント
- 情報セキュリティ
- 組み込みアプリケーション
- 算術および科学

マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上





インテル® IPP のパフォーマンス最適化

インテル® IPP 関数は命令セット・アーキテクチャーを活用

- ストリーミング SIMD 拡張命令 SSE、SSE2、SSE3、SSSE3、SSE4.1
- インテル® MMX® テクノロジー

インテル® IPP 関数はプロセッサのマイクロアーキテクチャーを活用

- データのプリフェッチおよびキャッシュブロックの回避
- データとトレース・キャッシュ・ミスの解決
- 分岐予測ミスの回避

インテル® IPP 関数は従来のプロセッサ上で利用可能なスレッドレベルの並列処理を活用

- マルチコア・テクノロジー
- ハイパースレッディング・テクノロジー

インテル® IPP のテクノロジーは
プロセッサのパフォーマンスを最大限に引き出す

マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上



ライブラリーの命名規約



- インテル® IPP 関数には、特定のプロセッサでの実行用に最適化された多くのバージョンがある
- ディスパッチ手法は、プロセッサの種類を自動的に判断し、ジャンプテーブルまたはプロセッサ固有のライブラリーを使用して、各関数の最適なバージョンが実行され、最高のパフォーマンスが達成されることを保証する

プラットフォーム	ID	最適化
IA-32 インテル® アーキテクチャー	px	任意のインテル® Pentium® プロセッサ - C 最適化
	a6	SSE 最適化 (インテル® Pentium® III プロセッサ)
	w7	SSE2 最適化 (インテル® Pentium® 4 プロセッサ、インテル® Xeon® プロセッサ、 インテル® Centrino® モバイル・テクノロジー・ベースのプロセッサ)
	t7	SSE3 最適化 (ハイパースレッディング・テクノロジー対応 インテル® Pentium® 4 プロセッサ)
	v8	SSSE3 最適化 (インテル® Core™2 Duo プロセッサ)
	p8	SSE4.1 最適化 (45nm インテル® Core™2 Duo (開発コード名 Penryn) プロセッサ・ ファミリー)
インテル® 64 (インテル® EM64T) アーキテクチャー	mx	すべてのインテル® 64 ベースのプラットフォーム - C 最適化
	m7	SSE3 最適化 (インテル® 64 ベースのアーキテクチャー)
	u8	SSSE3 最適化 (インテル® Core™2 Duo プロセッサ)
	y8	SSE4.1 最適化 (45nm インテル® Core™2 Duo (開発コード名 Penryn) プロセッサ・ ファミリー)
インテル® Itanium® アーキテクチャー	i7	インテル® Itanium® プロセッサ・ファミリー用に最適化
インテル® IXP4xx ネットワーク・ プロセッサ	sx	IXP4XX プロダクトライン向けの C 最適化
	s2	IXP4XX プロダクトライン向けの最適化

インテル® IPP
5.3 で追加された
最適化

インテル® IPP は最新のプロセッサ機能に対応した最新ライブラリーを使用

マルチスレッド・ライブラリー





インテル® IPP ライブラリーのコンポーネント

- ヘッダーファイル、およびダイナミック・ライブラリーとスタティック・ライブラリーはドメインでソートされる
- ディスパッチャー・ライブラリーと SSE ベースの最適化ライブラリーはダイナミック・ライブラリーとスタティック・ライブラリーの両方に含まれる

例:

ドメイン	ヘッダーファイル	ダイナミック・リンク	スタティック・リンク
画像処理 (i)	<i>ippi.h</i>	<i>ippi.lib</i> , <i>ippi-5.3.dll</i> , <i>ippip8-5.3.dll</i> , <i>ippiv8-5.3.dll</i> , <i>ippit7-5.3.dll</i> <i>ippiw7-5.3.dll</i> , <i>ippia6-5.3.dll</i> , <i>ippipx-5.3.dll</i>	<i>ippimerged.lib</i> , <i>ippiemerged.lib</i>

複数のリンクモデル
複数のヘッダーファイルによるライブラリー構造

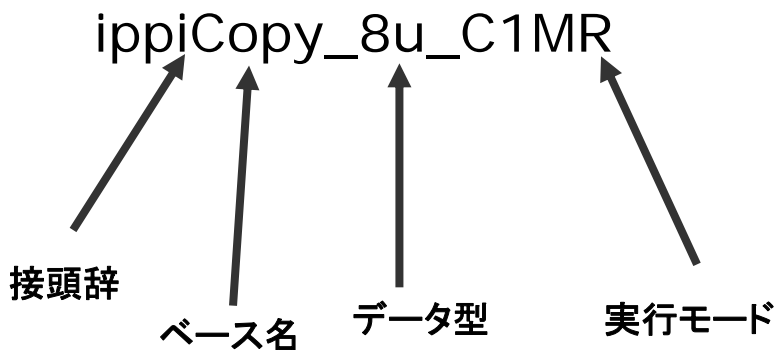
マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上



関数の命名規約

関数名は..

- 簡単に理解できる
- 明瞭な要素により、関数の目的を直感的に示す
- 各要素は定義済みの固定数の値を持つ



名前要素	説明	例
接頭辞	1D、2D、行列および 3D データ処理の関数データドメインを示す	ipps、ippi、ippm、ippr
ベース名	主となる操作の短縮形	Add、FFTFwd、LuDecomp
データ型	ビットの深さと符号を示す	8u、32f、64f...
実行モード	データレイアウトとスケーリングを示す	ISfs、C1R、P...

説明: 画像 (ippi) コピー (Copy) 操作。データ型は 8u (8u)。画像は 1 チャンネル (C1) でマスク (M) を特定の領域 (R) で使用する。

各関数は既知のデータ型により、特定のモードで特定の操作を実行

マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上



インテル® IPP がサポートしているリンクモデルの比較

リンクモデル 機能	1. ダイナミック・リンク	2. スタティック・リンク (ディスパッチあり)	3. カスタム・ ダイナミック・リンク	4. スタティック・リンク (ディスパッチなし)
最適化	すべてのプロセッサ	すべてのプロセッサ	すべてのプロセッサ	1 プロセッサ
ビルド	スタブ・ライブラリーに リンク	スタティック・ ライブラリーとスタブに リンク	個別の DLL をビルド	プロセッサ固有の マージ・ライブラリー にリンク
呼び出し	通常の名前	通常の名前	修正した名前	プロセッサ固有の 名前
合計バイナリーサイズ	大	小	小	最小
実行ファイルのサイズ	最小	小	最小	小
カーネルモード	×	○	×	○

インテル® IPP は多くの柔軟性を提供

マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上





インテル® IPP とマルチスレッディング

1. インテル® IPP におけるスレッド化
2. インテル® IPP でスレッド化したマルチメディア・サンプル・コード
3. インテル® IPP におけるスレッド化の安全性





インテル® IPP 関数におけるスレッド化

- 多くの計算集約関数が高スレッド化
- 約 1800 のインテル® IPP 関数が高スレッド化
- パフォーマンスが向上
- 使用モデル
- インテル® IPP スレッド化制御
 - *ippSetNumThreads*
 - *ippGetNumThreads*

ドメイン	スレッド化された関数の数
カラー変換	100
コンピューター・ビジョン	132
暗号化	98
画像処理	673
JPEG/JPEG 2000	36
行列	541
信号処理	215

インテル® IPP 関数は
スレッド化によりパフォーマンスが向上

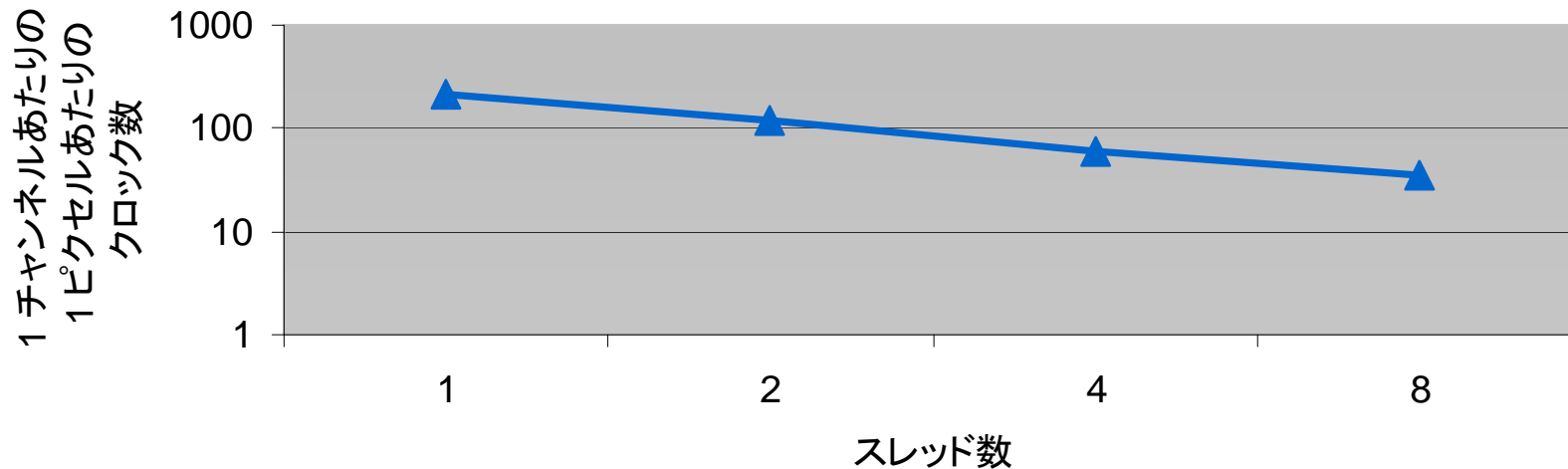
マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上



マルチコアシステムにおけるインテル® IPP 関数のパフォーマンス・データ



画像処理関数 `ippiCrossCorrValid_NormLevel_32f_C1R` のパフォーマンス
画像 = 1024x1024
検索する機能 = 64x64



実行ファイル: `ps_ippi.exe ps_ippm.exe -Nx スイッチ`

ライブラリー: インテル® IPP v5.3, `ippiv8-5.3.dll`, `ippmv8-5.3.dll` IA-32 インテル® C++ コンパイラー 10.0

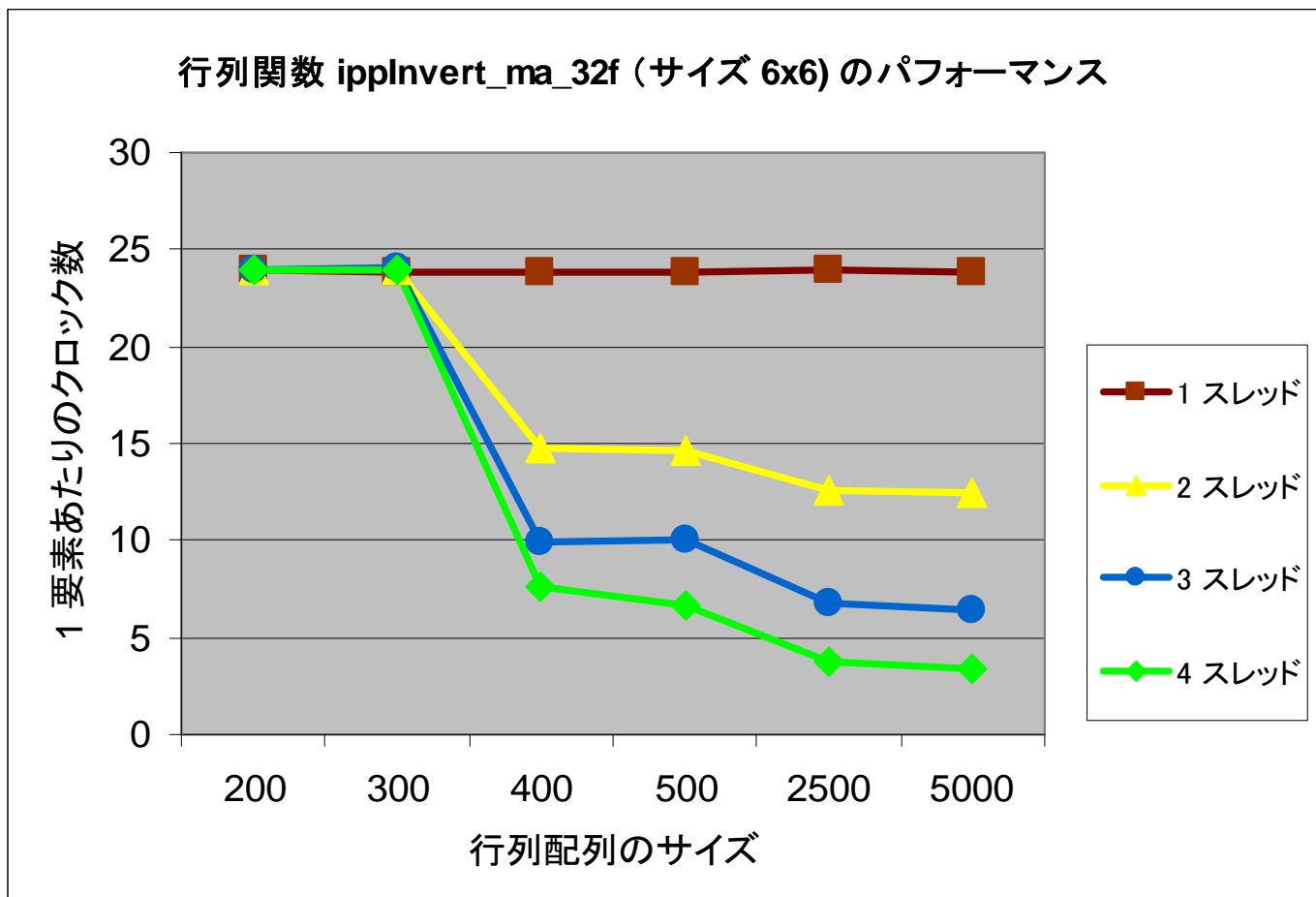
システム: インテル® Core™2 Quad プロセッサ 8x2400 MHz, L1=32/32K

OS: Microsoft® Windows Server® 2003 (Win32)

マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上



マルチコアシステムにおけるインテル® IPP 関数のパフォーマンス・データ



実行ファイル: `ps_ippi.exe ps_ippm.exe -Nx スイッチ`

ライブラリー: インテル® IPP v5.3, `ippiv8-5.3.dll`, `ippmv8-5.3.dll` IA-32 インテル® C++ コンパイラー 10.0

システム: インテル® Core™2 Quad プロセッサ 8x2400 MHz, L1=32/32K

OS: Microsoft® Windows Server® 2003 (Win32)

マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上



マルチメディア・アプリケーションの挑戦



- 新しい高精細デジタルメディア規格では、より高い圧縮率と画質が実現されるが、アルゴリズムもより複雑になる
 - ビデオ: H.264 と MPEG-2
 - 画像: JPEG 2000 と JPEG
 - オーディオ: AAC と MP3
- 新しいマルチコア・コンピューティング・プラットフォームでは、より複雑なマルチスレッド・プログラミングが必要になる
 - 低レベル操作のスレッド化だけでは十分でない
 - 計算カーネルは小規模のデータで動作するが、高レベルのアルゴリズムで制御される異なるデータを繰り返し呼び出す



インテル® IPP メディアコーデックにおけるスレッド化



コーデック	スレッディング・メカニズム	スレッディング・テクノロジー	パフォーマンス・ゲイン
H.264 デコーディング	ネイティブ・システム・スレッディング	<ul style="list-style-type: none"> スライスの並列化 – 単純なスレッド化モデル スライス内 (フレーム内) の並列化 フレーム間の並列化 	<ul style="list-style-type: none"> 2 スレッド (デュアルコア) で約 2 倍 - HD ストリーム (CPU 負荷 100%) 4 スレッド (クアッドコア) で約 3.2~3.7 倍 - HD ストリーム (CPU 負荷 100%)
VC-1 デコーディング	ネイティブ・システム・スレッディング	<ul style="list-style-type: none"> スライスの並列化 – 単純なスレッド化モデル スライス内 (フレーム内) の並列化 フレーム間の並列化 	<ul style="list-style-type: none"> 2 スレッド (デュアルコア) で約 1.85~1.9 倍 - HD ストリーム (CPU 負荷 100%) 4 スレッド (クアッドコア) で約 3.2~3.3 倍 - HD ストリーム (CPU 負荷 100%)
JPEG 2000 デコーディング	OpenMP*	<ul style="list-style-type: none"> 各コードブロックは並列で処理される 	<ul style="list-style-type: none"> 4 スレッド (クアッドコア) で約 2.5~3.1 倍 - 中サイズの画像 (CPU 負荷 100%)
JPEG デコーディング	OpenMP*	<ul style="list-style-type: none"> 各 MCU 行は並列で処理される 	<ul style="list-style-type: none"> 2 スレッド (デュアルコア) で約 1.9 倍 - 中サイズの画像 (CPU 負荷 100%)

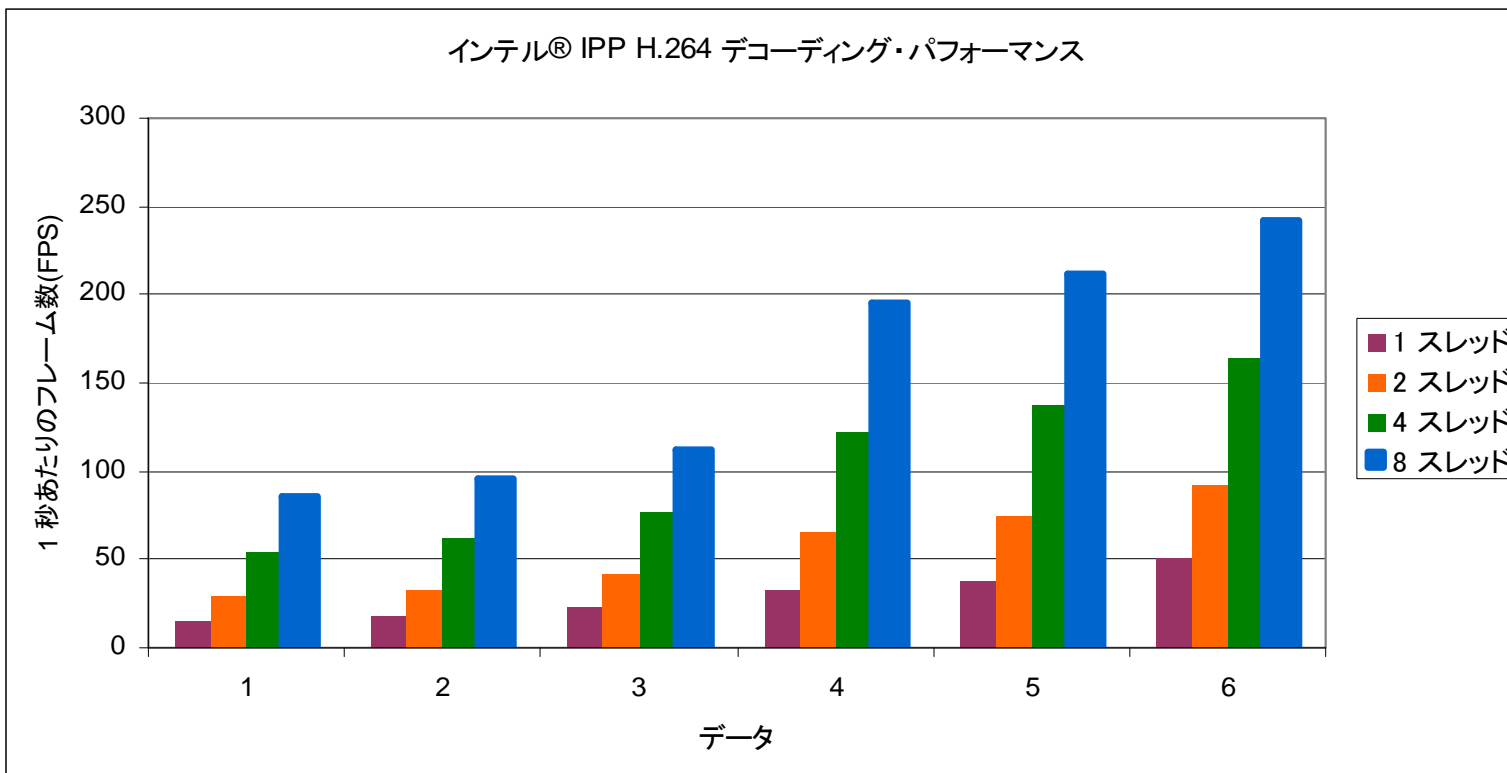
マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上



UMC H264 デコーダー・スケーリング



インテル® IPP H.264 デコーディング・パフォーマンス



2xインテル® Xeon® (開発コード名 Clovertown) プロセッサ 2.40GHz、Microsoft* Windows Server* 2003 Enterprise x64 Edition

インテル® C++ コンパイラ 10.1、インテル® IPP 5.3

データ 1: H264_1920x1080_walking_couple_25Hz_main_IBBP_f200_40M.h264

データ 2: H264_1920x1080_walking_couple_25Hz_main_IBBP_f200_29M.h264

データ 3: H264_1920x1080_walking_couple_25Hz_main_IBBP_f200_15M.h264

データ 4: H264_1280x720_ShttleStart_60Hz_main_IBBP_f200_40M.h264

データ 5: H264_1280x720_ShttleStart_60Hz_main_IBBP_f200_29M.h264

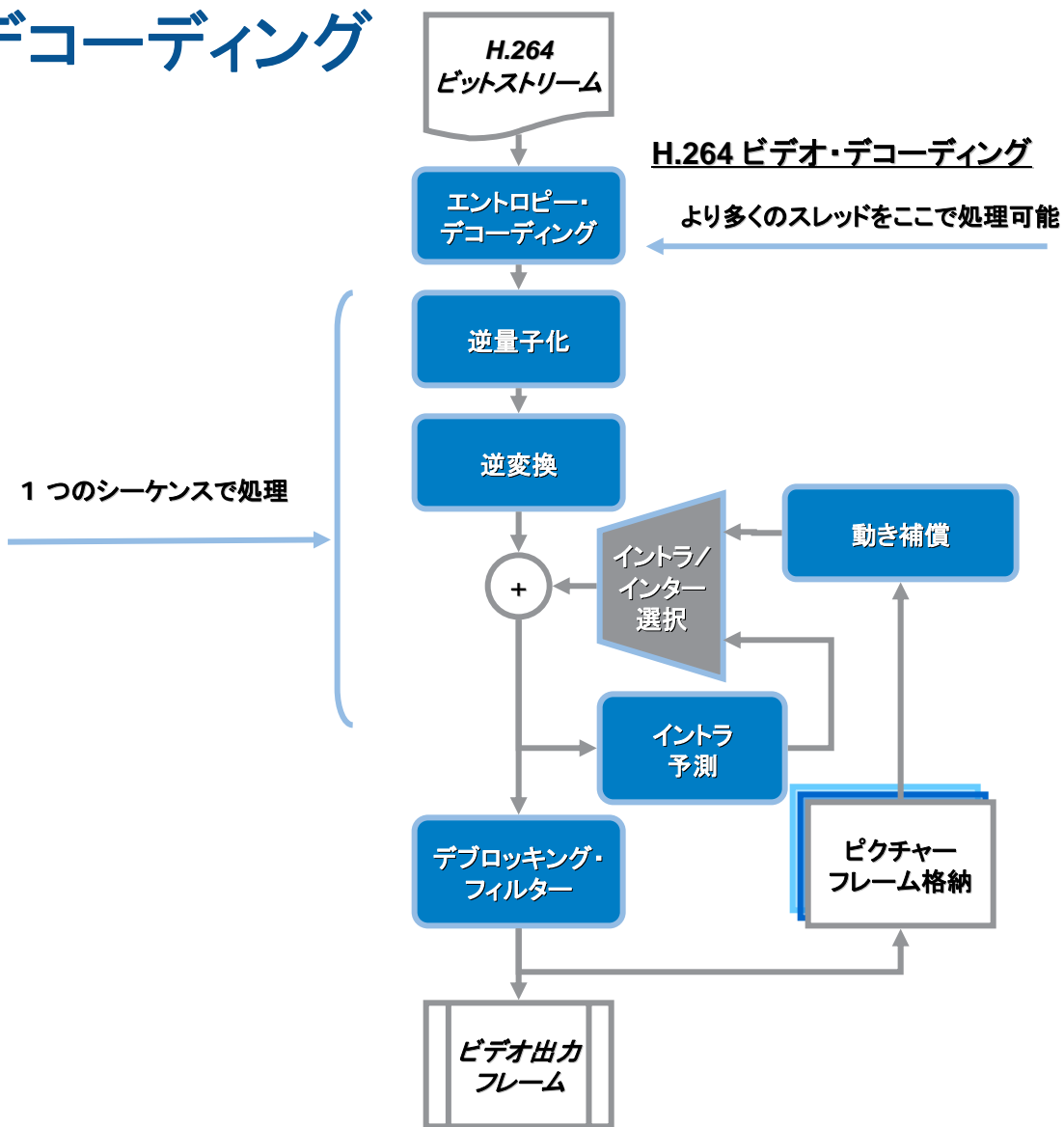
データ 6: H264_1280x720_ShttleStart_60Hz_main_IBBP_f200_15M.h264

H264 デコーダーはマルチコアで理論的な最大のスケーリングを達成

マルチス



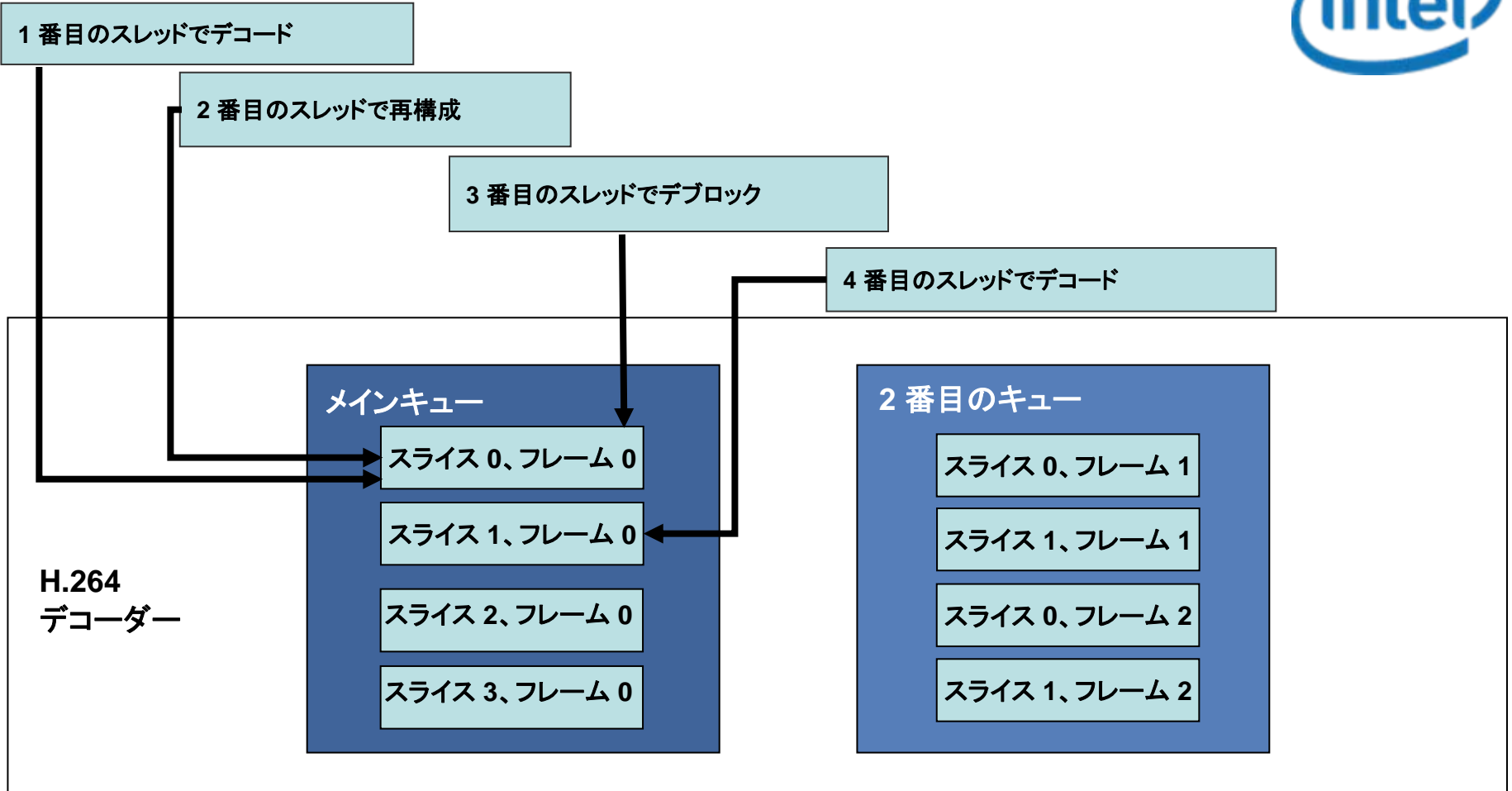
H.264 デコーディング



マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上



動作



メインキューの各スライスのデコード、再構成およびデブロックは並列で実行可能

マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上





スレッド・バランシング

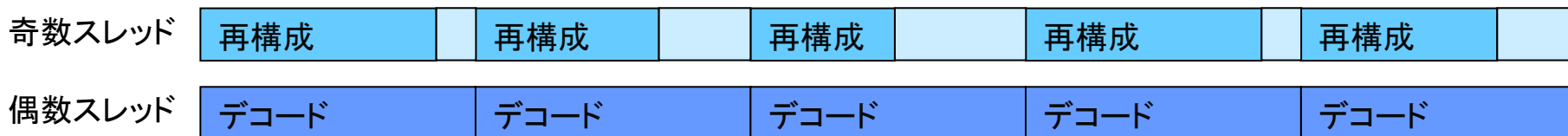
再構成がデコードよりも難しい場合


時間 →





再構成がデコードよりも簡単な場合

時間 →



 奇数スレッドで再構成を実行

 偶数スレッドでデコードを実行

 デブロッキングを実行

マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上



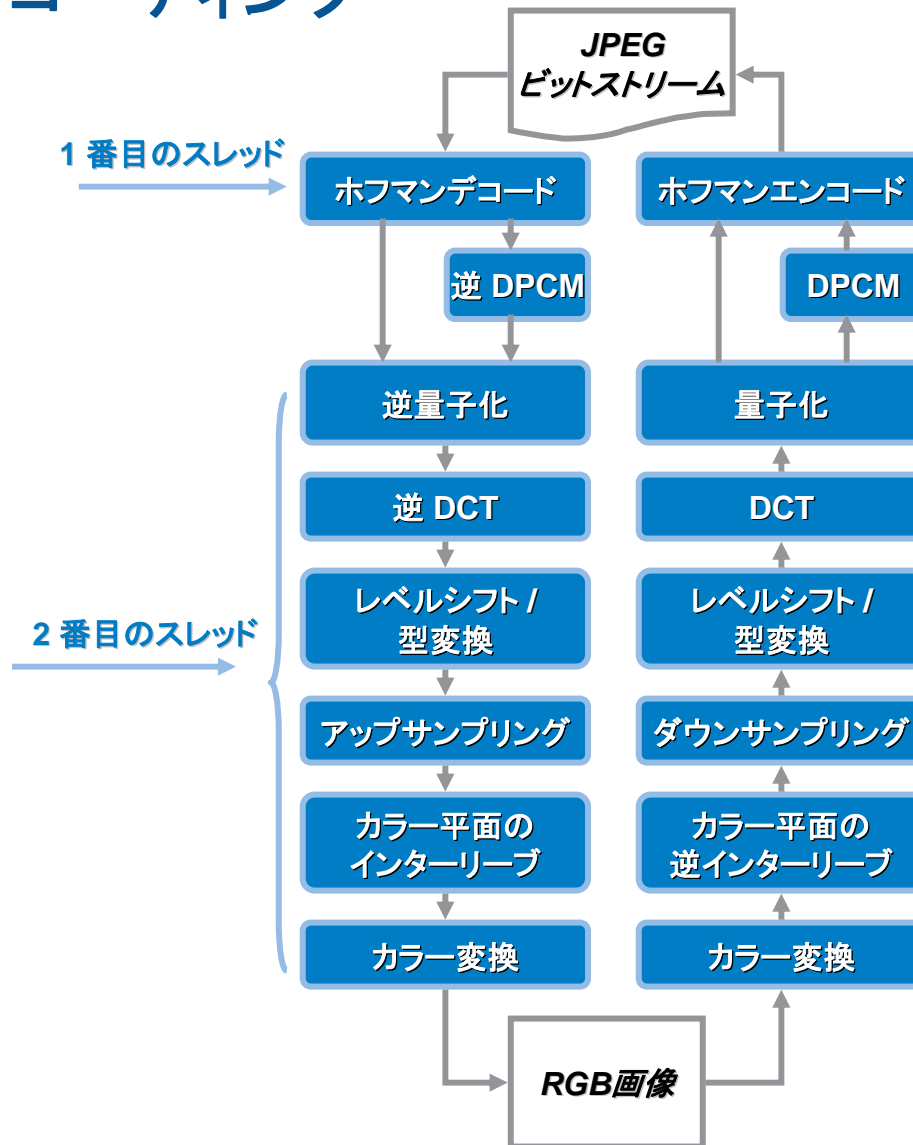
要約: インテル® IPP UMC スレッディング・ビデオ・コーディング



- インテル® IPP はマルチメディア・コーデックで優れたマルチコア・スケーラビリティを実現
- スレッディング・モデルは 8 スレッドまでスケーラブル
- 利用が簡単 - モデルは H264 と VC1 で同一。
任意の他のコーデックに適用可能



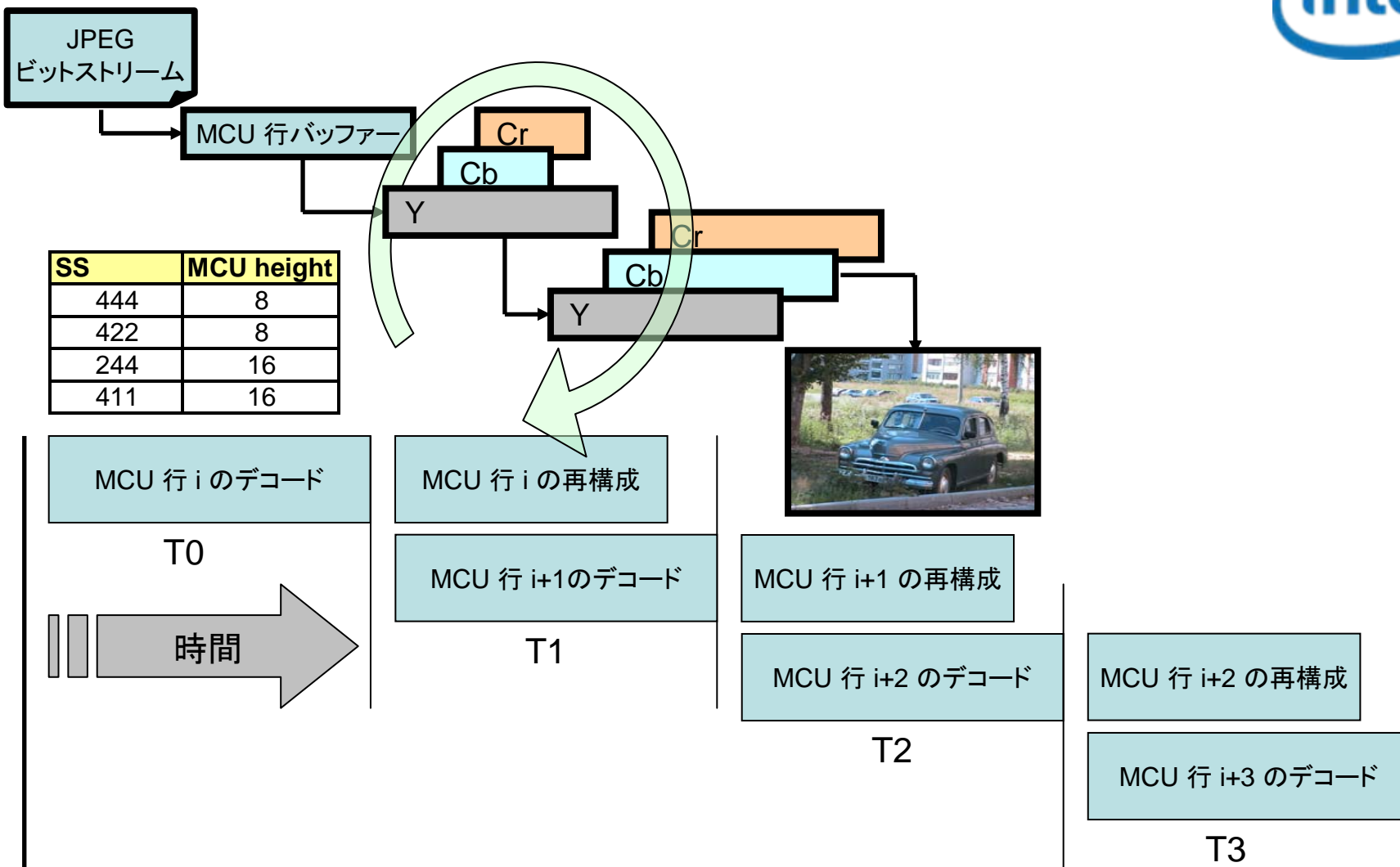
JPEG コーディング



マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上



JPEG デコーダー内部のスレッディング



マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上



JPEG および JPEG 2000 のパフォーマンス・データ

表 1: インテル® IPP ベースの JPEG デコーダー *

画像サイズ	サンプリング	パフォーマンス・ゲイン (1スレッド vs 2 スレッド)
	444	1.90
640x480	422	1.93
	411	1.93
	444	1.84
800x600	422	1.83
	411	1.76
	444	1.86
1024x768	422	1.81
	411	1.89
	444	1.89
4096x4096	422	1.82
	411	1.88

• インテル® IPP ベースの JPEG デコーダーは JPEG 画質パラメーター 75 の非可逆圧縮を使用します。

* 圧縮比は約 1:6 です。

表 2: インテル® IPP ベースの JPEG 2000 デコーダー

画像サイズ	wt 型	圧縮	パフォーマンス・ゲイン (1 スレッド vs 4 スレッド)
640x480	wt53	可逆	1.44
1024x768	wt53	可逆	1.66
4096x4096	wt53	可逆	1.55
640x480	wt97	15*	2.59
1024x768	wt97	15	2.64
4096x4096	wt97	15	3.36

• 「圧縮」列の値 15 は圧縮比 1:15 を意味します。

表 1 および 2 で使用したシステム/ソフトウェア構成:

システム: インテル® Core™2 Duo プロセッサ 2.66GHz、
1GB メモリー、4MB キャッシュ。

コンパイラー: インテル® C++ コンパイラー 9.1、インテル® IPP 5.3
ダイナミック・ライブラリー。

インテル® IPP 5.3 ベースの JPEG ビューアー・サンプルおよび JPEG
2000 コーディング・サンプル

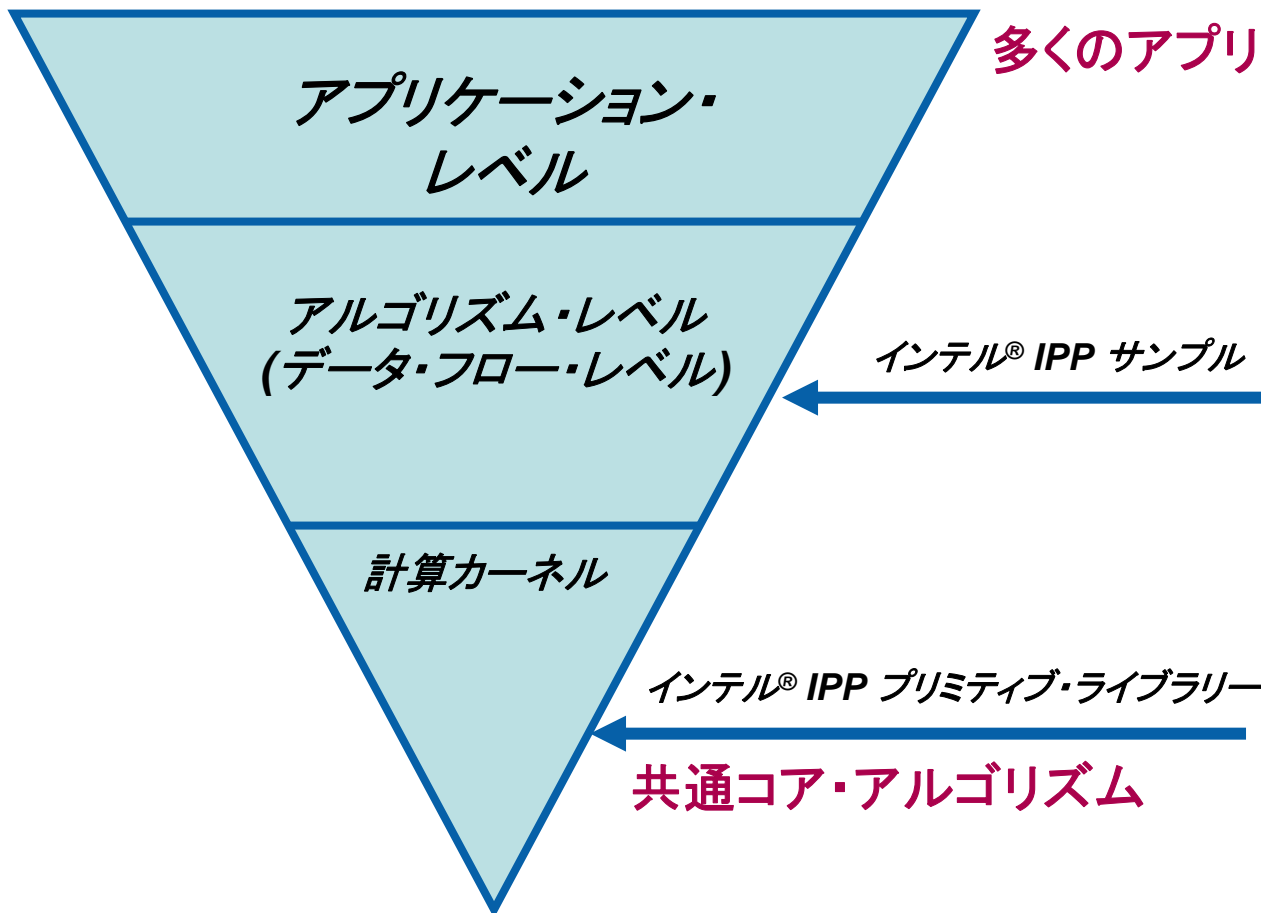
マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上



インテル® IPP におけるスレッド制御の柔軟性

- インテル® IPP はスレッドセーフでスレッド化をサポート
 - インテル® IPP スレッド化関数は独立しているため、アプリケーションレベルのスレッド化で OpenMP* を使用する必要がない
 - インテル® IPP のスレッド化はアプリケーションによって無効にするか、微調整が可能
- アプリケーションで細粒度のスレッド化制御が必要な場合
 - 引数 1 で関数 *ippSetNumThreads* を呼び出す
 - シングルスレッド・アプリケーションまたはカーネル開発用のスタティックライブラリーを使用する





多くのアプリケーション

- さまざまなレベルでスレッド化を最適化
- インテル® IPP サンプルでアルゴリズム・レベルのスレッド化を説明
- インテル® IPP プリミティブ・ライブラリーでスレッド化可能なデータをスレッド化



インテル® IPP 5.3

高度に最適化されたマルチメディア関数

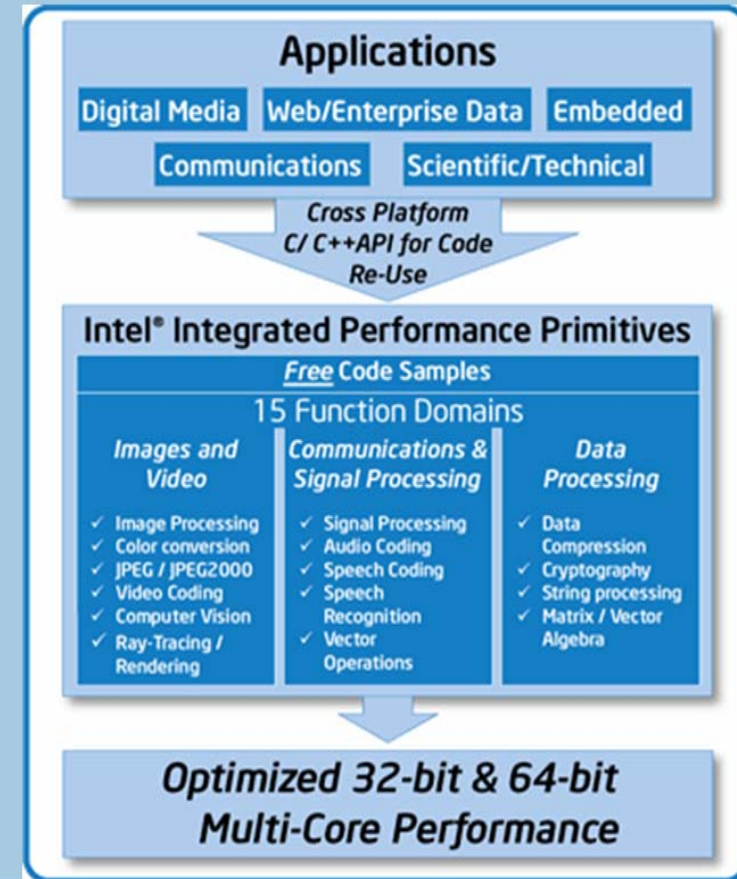
新機能

- 45nm インテル® Core™2 Duo (開発コード名 Penryn) プロセッサファミリー用の最適化
- gzip および bzip2 と完全に互換性のある新しいデータ圧縮関数とコードサンプル
- データ圧縮パフォーマンスの向上
- 新しいプロセッサ ID サンプル
- 新しい 3D サイズ変更関数、2D サイズ変更補間画像サンプル

Windows*	Linux*	Mac*	IA32	インテル® 64	IA64	マルチコア
√	√	√	√	√	√	√

We are impressed by the breadth and depth of Intel's software tools. After rigorous testing and evaluation of Intel Integrated Performance Primitives (Intel IPP), we are incorporating Intel IPP in our Premiere Pro* product. Adobe Premiere Pro software is a revolutionary nonlinear video editing application that takes video and audio production to an entirely new level, and Intel IPP has been a critical part of our success in delivering cutting-edge technology. Intel IPP will help us maintain our competitive edge."

Steve Warner
Engineering Manager
Adobe Premiere Pro
Adobe Systems, Inc.



マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上



45nm インテル® Core™2 Duo (開発コード名 Penryn) プロセッサにおけるインテル® IPP 関数のパフォーマンス



関数	パフォーマンス・ゲイン (5.3 p8/5.2 v8)	SSE4.1 命令
ippsSqrt_32f	14%	pptest
ippiConvert_8u32s_AC4R	13%	pblendv、 pmovsx/pmovzx
ippsRijndael256DecryptCFB_8u	24%	pinsrb/pextrb

システム: ストリーミング SIMD 拡張命令セット 4.1 2x2400 MHz

OS: Windows Vista* Business (Win32)

インテル® IPP 5.3/5.2、ダイナミック・ライブラリー

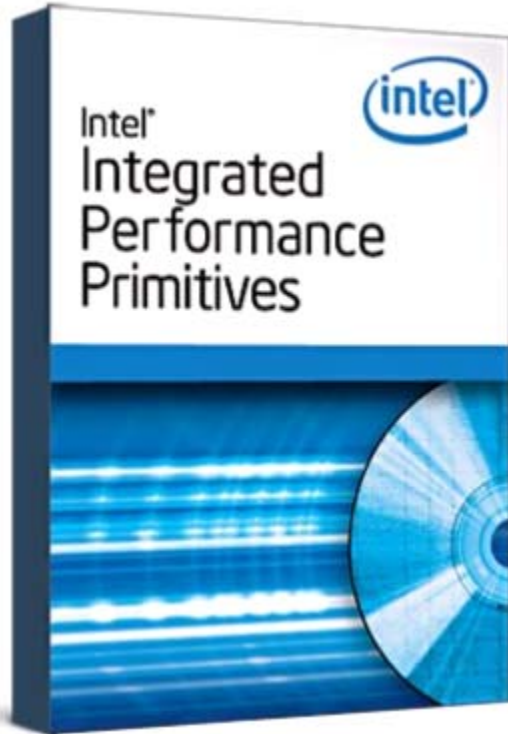
インテル® C++ コンパイラー 10.0

マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上





要約



IPP Benefits	
✓	Boost application performance
✓	Thousands of hand optimized funtions
✓	Free downloadable Code Samples

すべてのインテル® ソフトウェア開発製品は評価版をダウンロードできます。
<http://www.intel.co.jp/jp/software/products/>

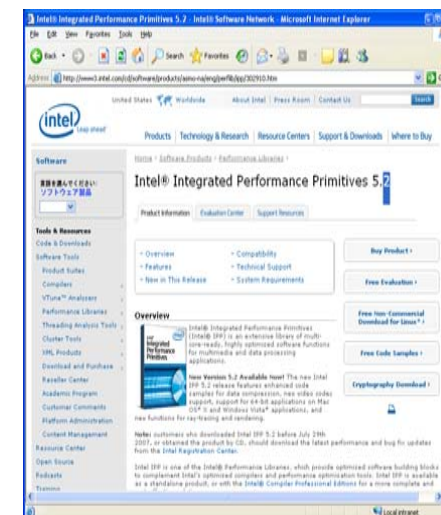
マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上





インテル® IPP 製品リファレンス

- インテル® IPP 5.3 Web サイト
(製品リリース、パフォーマンス・データ、サポートリソース)
<http://www.intel.com/software/products/ipp>
- インテル® IPP ブック (2nd Edition)
<http://shop.intel.com/shop/product.aspx?pid=SIBK3608>
- インテル® IPP フォーラム
<http://softwarecommunity.intel.com/isn/Community/en-US/forums/1274/ShowForum.aspx>
- インテル® プレミアサポート (テクニカルサポート)
<https://premier.intel.com>



マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上





公開中の Web セミナー

現在公開中の技術 Web セミナー シリーズ
並行処理の革命 (近日公開)

<http://event.on24.com/event/36/88/3/rt/1/> ページで [Japanese] を選択してアクセスできます

1. パラレル・プログラミングが主流に: 準備はできていますか?
2. パラレル・ソフトウェアの基本紹介
3. マルチコア CPU と Windows Vista* 向けのソフトウェア・パフォーマンス分析スレッディング、パフォーマンスへの 3 つのステップ
4. Part 1 - スレッドの正当性: スレッド化ソフトウェアの開発、管理、チューニングにおいて決定論的な結果を維持する
5. Part 2 - 並列処理を明示する: インテル® スレッディング・ビルディング・ブロックの事例
6. Part 3 - スレッド化ソフトウェアをチューニングする: 並列化の後のステップ
7. パフォーマンス、マルチスレッディング、セキュリティのためにインテル® C++/Fortran コンパイラ バージョン 10.0 を使用する

マルチスレッド・ライブラリーを使用したデジタル・メディア・アプリケーションのパフォーマンスの向上





www.intel.co.jp/jp/software/products/

インテル® IPP ライブラリー



ドメイン	ヘッダーファイル	ダイナミック・リンク	スタティック・リンク
オーディオ・コーディング	ippac.h	ippac.lib	ippacmerged.lib
ビデオ・コーディング	ippvc.h	ippvc.lib	ippvcmerged.lib
カラー変換	ippcc.h	ippcc.lib	ippccmerged.lib
文字列処理	ippch.h	ippch.lib	ippchmerged.lib
暗号化	ippcp.h	ippcp.lib	ippcpmerged.lib
コンピューター・ビジョン	ippcv.h	ippcv.lib	ippcvmerged.lib
データ圧縮	ippdc.h	ippdc.lib	ippdcmerged.lib
画像処理	ippi.h	ippi.lib	ippimerged.lib
JPEG 処理	ippj.h	ippj.lib	ippjmerged.lib
行列演算	ippm.h	ippm.lib	ippmmerged.lib
信号処理	ipps.h	ipps.lib	ippsmerged.lib
音声コーディング	ippsc.h	ippsc.lib	ippscmerged.lib
音声認識	ippsr.h	ippsr.lib	ippsrmerged.lib
ベクトル演算	ippvm.h	ippvm.lib	ippvmmerged.lib
リアリスティック・レンダリング	ippr.h	ippr.lib	ipprmerged.lib
共通関数	ippcore.h	ippcore.lib	ippcorel.lib

複数のリンクモデル
複数のヘッダーファイルによるライブラリー構造

マルチスレッド・ライブラリーを使用したデジタル・メディア



本資料に掲載されている情報は、インテル製品の概要説明を目的としたものです。製品に付属の売買契約書『Intel's Terms and conditions of Sales』に規定されている場合を除き、インテルはいかなる責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、知的所有権を侵害していないことへの保証を含む）に関しても一切責任を負わないものとします。

インテル製品は、予告なく仕様が変更されることがあります。

* 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。

© 2008 Intel Corporation.