

# インテル® C++ コンパイラー 11.1 Linux\* 版 プロフェッショナル・エディション インストール・ガイドおよびリリースノート

---

資料番号: 321412-002JA

2010年3月24日

## 目次

1	概要.....	4
1.1	変更履歴.....	4
1.2	製品の内容.....	5
1.3	動作環境.....	5
1.3.1	Red Hat Enterprise Linux 3、SUSE LINUX Enterprise Server 9 のサポート 終了予定.....	7
1.3.2	クラスター OpenMP ライブラリーのコマンドライン・オプションの廃止 予定.....	7
1.4	ドキュメント.....	7
1.5	日本語サポート.....	8
1.6	テクニカルサポート.....	8
2	インストール.....	8
2.1.1	サイレントインストール.....	8
2.1.2	Eclipse 統合のインストール.....	9
2.1.3	既知のインストールの問題.....	9
2.2	インストール先フォルダー.....	9
2.3	削除/アンインストール.....	11
3	インテル® C++ コンパイラー.....	11
3.1	互換性.....	11
3.2	新機能と変更された機能.....	11
3.3	新規および変更されたコンパイラー・オプション.....	11
3.3.1	-O0 オプションの -mp の除外.....	12
3.4	その他の変更.....	12
3.4.1	最適化レポートがデフォルトで無効に設定.....	12
3.4.2	コンパイラー環境の構築.....	12
3.4.3	デフォルトの命令セットがインテル® ストリーミング SIMD 拡張命令 2 (インテル® SSE2) を必要とするものに変更.....	12
3.4.4	OpenMP ライブラリーのデフォルトが "compat" に変更.....	13

3.4.5	mathf.h の削除.....	13
3.4.6	サンプリング・ベースのプロファイルに基づく最適化機能の削除.....	13
3.4.7	OpenMP ヘッダーファイルの変更 .....	13
3.5	Eclipse IDE でのソースチェッカーの使用.....	14
3.6	既知の問題.....	14
3.6.1	TR1 システムヘッダー.....	14
3.6.2	KMP_AFFINITY のデフォルト動作の変更.....	15
3.6.3	古いバージョンの ld による致命的なエラー .....	15
3.6.4	一部の Intel® AVX アーキテクチャー命令と組み込み命令の削除.....	15
4	Intel® デバッガー (IDB).....	15
4.1	Java ランタイム環境の設定.....	15
4.2	デバッガーの起動.....	16
4.3	その他のドキュメント.....	16
4.4	デバッガー機能.....	16
4.4.1	IDB の主な機能.....	16
4.4.2	新機能と変更された機能.....	16
4.5	既知の問題.....	17
4.5.1	データ共有検出の問題.....	17
4.5.2	[Signals (シグナル)] ダイアログが動作しない .....	17
4.5.3	再入可能呼び出しにおけるブレークの未サポート .....	17
4.5.4	GUI のサイズ調整 .....	17
4.5.5	プロセスの終了.....	17
4.5.6	10 進浮動小数点の未サポート.....	17
4.5.7	\$cdir ディレクトリー、\$cwd ディレクトリー.....	17
4.5.8	info stack の使用.....	18
4.5.9	\$stepg0 のデフォルト値の変更.....	18
4.5.10	一部の Linux システムでの SIGTRAP エラー .....	18
4.5.11	MPI プロセスのデバッグには idb GUI は使用できません.....	18
5	Eclipse 統合.....	18
5.1	提供されている統合 .....	19
5.1.1	Eclipse 3.5 および CDT 6.0.....	19
5.1.2	Eclipse 3.4 および CDT 5.0.....	19
5.1.3	統合に関する注意事項.....	19
5.2	Eclipse での Intel® C++ Eclipse 製品拡張のインストール方法.....	19
5.2.1	Eclipse 3.5.0 および CDT 6.0.0 "Galileo" .....	19
5.2.2	Eclipse 3.4.0 および CDT 5.0.0 "Ganymede" .....	20

5.3	Eclipse、CDT、および JRE の入手方法とインストール方法 .....	21
5.3.1	Eclipse 3.5.0 および CDT 6.0.0.....	21
5.3.2	Eclipse 3.4.0 および CDT 5.0.0.....	21
5.3.3	JRE、Eclipse、CDT のインストール.....	21
5.4	インテル® C++ コンパイラーで開発するための Eclipse の起動.....	22
5.5	Fedora システムでのインストール.....	22
5.6	コンパイラー・バージョンの選択.....	22
6	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 .....	23
6.1	新機能と変更された機能.....	23
6.1.1	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 6.1 Update 5.....	23
6.1.2	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 6.1 Update 4.....	23
6.1.3	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 6.1 Update 3.....	23
6.1.4	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 6.1 Update 2.....	23
6.2	既知の制限事項.....	24
6.3	別途ダウンロード可能なインテル® IPP 暗号化ライブラリー .....	24
6.4	インテル® IPP コードサンプル .....	24
7	インテル® マス・カーネル・ライブラリー.....	24
7.1	本バージョンでの変更.....	24
7.1.1	インテル® マス・カーネル・ライブラリー 10.2 Update 5 .....	24
7.1.2	インテル® マス・カーネル・ライブラリー 10.2 Update 4 .....	25
7.1.3	インテル® マス・カーネル・ライブラリー 10.2 Update 3 .....	25
7.1.4	インテル® マス・カーネル・ライブラリー 10.2 Update 2 .....	26
7.2	既知の問題.....	28
7.3	注意事項.....	28
7.4	権利の帰属.....	28
8	インテル® スレッディング・ビルディング・ブロック .....	28
8.1	本バージョンでの変更.....	29
8.1.1	インテル® スレッディング・ビルディング・ブロック 2.2 Update 2.....	29
8.1.2	インテル® スレッディング・ビルディング・ブロック 2.2 Update 1.....	29
8.2	既知の問題.....	30
9	著作権と商標について.....	31

# 1 概要

このドキュメントでは、製品のインストール方法、新機能、変更された機能、注意事項、および製品ドキュメントに記述されていない既知の問題について説明します。

## 1.1 変更履歴

このセクションでは製品アップデートにおける重要な変更内容を説明します。報告されている問題の修正リストは、[インテル® コンパイラー 11.1 プロフェッショナル・エディション 修正リスト](#) (英語)、[インテル® IPP ライブラリー 6.1 修正リスト](#) (英語)、[インテル® マス・カーネル・ライブラリー 10.2 修正リスト](#) (英語) および [インテル® スレディング・ビルディング・ブロック 2.2 の変更](#) を参照してください。

### Update 6

- [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](#)が 6.1 Update 5 に更新
- [インテル® マス・カーネル・ライブラリー](#)が 10.2 Update 5 に更新
- 報告されている問題の修正

### Update 5 (11.1.069)

- [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](#)が 6.1 Update 4 に更新
- [インテル® マス・カーネル・ライブラリー](#)が 10.2 Update 4 に更新
- [インテル® スレディング・ビルディング・ブロック](#)が 2.2 Update 2 に更新
- [クラスター OpenMP\\* ライブラリーの廃止予定](#)に関する注意事項
- 報告されている問題の修正

### Update 4 (11.1.064)

- [OpenMP ヘッダーファイルの変更](#)によりエラー検出が向上
- [インテル® マス・カーネル・ライブラリー](#)が 10.2 Update 3 に更新
- [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](#)が 6.1 Update 3 に更新
- 報告されている問題の修正

### Update 3 (11.1.059)

- [インテル® スレディング・ビルディング・ブロック](#)が 2.2 Update 1 に更新
- 報告されている問題の修正

### Update 2 (11.1.056)

- Ubuntu\* 9.04 のサポートの追加
- [Fedora\\* 10 での非 RPM インストール](#)に関する注意事項の追加
- [新しいオプション -mk1 と -xAVX](#)に関する注意事項の追加
- [一部のインテル® AVX アーキテクチャー命令と組み込み命令の削除](#)に関する注意事項の追加
- Eclipse\* CDT 6.0 のフルサポート
- 報告されている問題の修正

## Update 1 (11.1.046)

- [-00 の動作変更](#)に関する注意事項の追加
- Eclipse CDT 6.0 の部分サポートの追加
- 報告されている問題の修正

## 1.2 製品の内容

インテル® C++ コンパイラー 11.1 Linux 版プロフェッショナル・エディションには、次のコンポーネントが含まれています。

- インテル® C++ コンパイラー。Linux オペレーティング・システムを実行する IA-32、インテル® 64、および IA-64 アーキテクチャー・システムで動作するアプリケーションをビルドします。
- インテル® デバッガー
- IA-64 対応アプリケーション開発用インテル® アセンブラー
- 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 6.1 Update 5
- インテル® マス・カーネル・ライブラリー 10.2 Update 5
- インテル® スレディング・ビルディング・ブロック 2.2 Update 2
- Eclipse 開発環境への統合
- 各種ドキュメント

## 1.3 動作環境

アーキテクチャー名についての説明は、<http://software.intel.com/en-us/articles/intel-architecture-platform-terminology/> (英語) を参照してください。

### IA-32 対応アプリケーション開発に必要な環境

- インテル® ストリーミング SIMD 拡張命令 2 (インテル® SSE2) 対応の IA-32 またはインテル® 64 アーキテクチャー・プロセッサをベースとするコンピューター (インテル® Pentium® 4 プロセッサ以降、または互換性のあるインテル以外のプロセッサ)
  - ホストと異なるターゲットの開発を行う場合、Linux ディストリビューションから別のライブラリー・コンポーネントのインストールが必要になることがあります。
  - 機能を最大限に活用できるように、マルチコアまたはマルチプロセッサ・システムの使用を推奨します。
- RAM 1GB (2GB 推奨)
- 3GB のディスク空き容量 (すべての機能をインストールする場合)
- 次の Linux ディストリビューションのいずれか (本リストは、インテル社により動作確認が行われたディストリビューションのリストです。その他のディストリビューションでも動作する可能性はありますが、推奨しません。サポート情報につきましては、[「テクニカルサポート」](#) セクションを参照してください。)
  - Asianux\* 3.0
  - Debian\* 4.0
  - Fedora 10
  - Red Hat\* Enterprise Linux\* 3、4、5
  - SUSE LINUX Enterprise Server\* 9, 10, 11
  - TurboLinux\* 11
  - Ubuntu 9.04
- Linux 開発ツール・コンポーネント (gcc、g++ および関連ツールを含む)
- binutils 2.17.50 以降
- libstdc++.so.5 を提供する Linux コンポーネント compat-libstdc++

- インテル® 64 アーキテクチャー・システムで開発を行う場合、一部の Linux ディストリビューションでは、次のいずれかまたは複数の Linux コンポーネントを追加でインストールしなければならない場合があります: ia32-libs、lib32gcc1、lib32stdc++6、libc6-dev-i386、gcc-multilib。

### インテル® 64 対応アプリケーションの開発に必要な環境

- インテル® 64 アーキテクチャー・プロセッサをベースとするコンピューター (インテル® Pentium® 4 プロセッサ以降、または互換性のあるインテル以外のプロセッサ)
  - 機能を最大限に活用できるように、マルチコアまたはマルチプロセッサ・システムの使用を推奨します。
- RAM 1GB (2GB 推奨)
- 3GB のディスク空き容量 (すべての機能をインストールする場合)
- 仮想メモリのページングファイル用に 100MB のディスク空き容量。インストールされている Linux のディストリビューションで推奨される最小容量以上の仮想メモリーを使用していることを確認してください。
- 次の Linux ディストリビューションのいずれか (本リストは、インテル社により動作確認が行われたディストリビューションのリストです。その他のディストリビューションでも動作する可能性はありますが、推奨しません。サポート情報につきましては、「[テクニカルサポート](#)」セクションを参照してください。)
  - Asianux 3.0
  - Debian 4.0
  - Fedora 10
  - Red Hat Enterprise Linux 3、4、5
  - SUSE LINUX Enterprise Server 9、10、11
  - TurboLinux 11
  - Ubuntu 9.04
- Linux 開発ツール・コンポーネント (gcc、g++ および関連ツールを含む)
- binutils 2.17.50 以降
- libstdc++.so.5 を提供する Linux コンポーネント compat-libstdc++
- 32 ビット・ライブラリーを含む Linux コンポーネント (ia32-libs と呼ばれる)

### IA-64 対応アプリケーション開発に必要な環境

- IA-64 アーキテクチャー・プロセッサ (インテル® Itanium® プロセッサ) をベースとするシステム
- RAM 1GB (2GB 推奨)
- 3GB のディスク空き容量 (すべての機能をインストールする場合)
- 次の Linux ディストリビューションのいずれか (本リストは、インテル社により動作確認が行われたディストリビューションのリストです。その他のディストリビューションでも動作する可能性はありますが、推奨しません。サポート情報につきましては、「[テクニカルサポート](#)」セクションを参照してください。)
  - Asianux 3.0
  - Debian 4.0
  - Red Hat Enterprise Linux 3, 4, 5
  - SUSE LINUX Enterprise Server 9、10、11
  - TurboLinux 11
  - Ubuntu 9.04
- Linux 開発ツール・コンポーネント (gcc、g++ および関連ツールを含む)
- binutils 2.17.50 以降
- libstdc++.so.5 を提供する Linux コンポーネント compat-libstdc++

## インテル® デバッガーのグラフィカル・ユーザー・インターフェイスを使用するためのその他の要件

- IA-32 アーキテクチャー・システムまたはインテル® 64 アーキテクチャー・システム
- Java® ランタイム環境 (JRE) 5.0 (1.5) または 6.0 (1.6)
  - IA-32 アーキテクチャー・システムでは 32 ビット版の JRE、インテル® 64 アーキテクチャー・システムでは 64 ビット版の JRE を使用する必要があります。

## Eclipse 統合を使用するためのその他の要件

- IA-32 アーキテクチャー・システムまたはインテル® 64 アーキテクチャー・システム
- Eclipse 3.5.x または 3.4.x
- Eclipse C/C++ Development Tools (CDT) 6.0.x または 5.0.x
- Java ランタイム環境 5.0 (1.5) または 6.0 (1.6)

## 説明

- インテル® コンパイラーは、さまざまな Linux ディストリビューションと gcc バージョンで動作確認されています。一部の Linux ディストリビューションには、動作確認に使用したヘッダーファイルとは異なるバージョンのものが含まれていて、問題を引き起こすことがあります。使用する glibc のバージョンは、gcc のバージョンと同じでなければなりません。最良の結果を得るため、上記のディストリビューションで提供されている gcc バージョンのみを使用してください。
- 非常に大きなソースファイル (数千行以上) を `-O3`、`-ipo` および `-openmp` などの高度な最適化オプションを使用してコンパイルする場合は、相当量の RAM が必要になります。
- 上記のリストにはすべてのプロセッサ・モデル名は含まれていません。リストされているプロセッサと同じ命令セットを正しくサポートしているプロセッサ・モデルでも動作します。特定のプロセッサ・モデルについては、「[テクニカルサポート](#)」セクションを参照してください。
- 一部の最適化オプションには、アプリケーションを実行するプロセッサの種類に関する制限があります。詳細は、オプションの説明を参照してください。

### 1.3.1 Red Hat Enterprise Linux 3、SUSE LINUX Enterprise Server 9 のサポート終了予定

インテル® C++ コンパイラーの将来のメジャーリリースでは、Red Hat Enterprise Linux 3 と SUSE LINUX Enterprise Server 9 はサポートされなくなる予定です。これらのオペレーティング・システムを使用している場合は、インテルでは新しいバージョンへの移行を推奨しています。

### 1.3.2 クラスタ OpenMP ライブラリーのコマンドライン・オプションの廃止予定

非公式なオプションとして [whatif.intel.com](http://whatif.intel.com) から利用可能なクラスタ OpenMP 機能は、コンパイラー製品のライブラリーとコマンドライン・サポートに依存しています。これらのライブラリーとコマンドライン・サポートは、コンパイラーの次のメジャーリリースで廃止されるため、クラスタ OpenMP 機能は使用できなくなります。

## 1.4 ドキュメント

製品ドキュメントは、「[インストール先フォルダー](#)」で示されているように、Documentation フォルダーに保存されています。

## 1.5 日本語サポート

インテル® コンパイラーは、日本語ユーザー向けのサポートを提供しています。エラーメッセージ、ビジュアル開発環境ダイアログ、ドキュメントの一部が英語のほかに日本語でも提供されています。エラーメッセージやダイアログの言語は、システムの言語設定に依存します。日本語版ドキュメントは、Documentation および Samples ディレクトリー以下の ja\_JP サブディレクトリーにあります。

日本語の言語サポートを英語のオペレーティング・システムで使用する場合や日本語のオペレーティング・システムで英語の言語サポートを使用する場合は、<http://software.intel.com/en-us/articles/changing-language-setting-to-see-english-on-a-japanese-os-environment-or-vice-versa-on-linux/> (英語) の説明を参照してください。

## 1.6 テクニカルサポート

[インテル® ソフトウェア開発製品レジストレーション・センター](#)でライセンスを登録してください。登録を行うことで、サポートサービス期間中 (通常は 1 年間)、製品アップデートと新しいバージョンの入手を含む無償テクニカルサポートが提供されます。

テクニカルサポート、製品のアップデート、ユーザーフォーラム、FAQ、ヒント、およびその他のサポート情報は、<http://www.intel.com/software/products/support/> (英語) を参照してください。

**注:** 代理店がテクニカルサポートを提供している場合は、インテルではなく代理店にお問い合わせください。

## 2 インストール

初めて製品をインストールする場合は、インストール中にシリアル番号の入力が求められますので、あらかじめご用意ください。製品のインストールと使用には、有効なライセンスが必要です。

DVD 版を購入した場合は、DVD をドライブに挿入し、DVD のトップレベル・ディレクトリーにディレクトリーを変更 (cd) して、次のコマンドでインストールを開始します。

```
./install.sh
```

ダウンロード版を購入した場合は、次のコマンドを使用して、書き込み可能な任意のディレクトリーに展開します。

```
tar -xzvf name-of-downloaded-file
```

その後、展開したファイルを含むディレクトリーに移動 (cd) し、次のコマンドでインストールを開始します。

```
./install.sh
```

手順に従ってインストールを完了します。

### 2.1.1 サイレントインストール

自動インストール、「サイレント」インストール機能についての詳細は、<http://software.intel.com/en-us/articles/intel-compilers-for-linux-version-111-silent-installation-guide/> (英語) を参照してください。

## 2.1.2 Eclipse 統合のインストール

「[Eclipse 統合](#)」セクションを参照してください。

## 2.1.3 既知のインストールの問題

- Linux ディストリビューションの Security-Enhanced Linux (SELinux) 機能を有効にしている場合は、インテル® C++ コンパイラーをインストールする前に SELINUX モードを permissive に変更する必要があります。詳細は、Linux ディストリビューションのドキュメントを参照してください。インストールが完了したら、SELINUX モードを元の値に戻してください。
- 一部の Linux バージョンでは、自動マウントデバイスに“実行”許可がなく、インストール・スクリプトを直接 DVD から実行すると、次のようなエラーメッセージが表示されることがあります。

```
bash: ./install.sh: /bin/bash: bad interpreter: Permission denied
```

このエラーが表示された場合は、次の例のように実行許可を含めて DVD を再マウントします。

```
mount /media/<dvd_label> -o remount,exec
```

その後、再度インストールを行ってください。

- Fedora 10 の一部のバージョンには、問題のある rpm ユーティリティーが含まれており、インテル® コンパイラーが正常にインストールできません。そのため、Fedora 10 システムでは、コンパイラーは非 RPM インストールを強制的に行います。
- バージョン 11.1 では、IA-32 およびインテル® 64 アーキテクチャー・システム上で Ubuntu 9.04 をサポートしています。ただし、ソフトウェアのライセンス規約上、Ubuntu 9.04 を実行しているインテル® 64 アーキテクチャー・システム上で、IA-32 コンポーネントを評価する際に、評価ライセンス機能を使用することはできません。Ubuntu の以前のバージョン (本リリースでは未サポート) でも同様の問題が発生する可能性があります。これは、評価ライセンス機能を使用する場合のみの問題です。シリアル番号、ライセンスファイル、フローティング・ライセンス、その他のライセンス・マネージャー操作、およびオフラインでのアクティベーション操作 (シリアル番号を使用) には、影響はありません。Ubuntu を実行しているインテル® 64 アーキテクチャー・システムで、バージョン 11.1 の IA-32 コンポーネントの評価が必要な場合は、インテル® ソフトウェア評価センター (<http://www.intel.com/cd/software/products/asmo-na/eng/download/eval/> (英語)) で評価版のシリアル番号を入手してください。

## 2.2 インストール先フォルダー

インストール・フォルダーの構成を以下に示します。一部含まれていないフォルダーもあります。

- <install-dir>/Compiler/11.1/xxx/
  - bin
    - ia32
    - intel64
    - ia64
  - include
    - ia32
    - intel64
    - ia64

- o perf\_headers
- o substitute\_headers
- o lib
  - ia32
  - intel64
  - ia64
- o eclipse\_support
- o idb
  - eclipse\_support
  - gui
  - ia32
  - ia64
  - intel64
  - lib
  - third\_party
- o ipp
  - em64t
  - ia32
  - ia64
- o mkl
  - benchmarks
  - examples
  - include
  - interfaces
  - lib
  - tests
  - tools
- o tbb
  - bin
  - em64t
  - examples
  - ia32
  - include
  - itanium
  - lib
- o Documentation
- o man
- o Samples

<install-dir> はインストール・ディレクトリー (デフォルトのインストール先は /opt/intel) で、xxx は 3 桁のビルド番号です。bin、include および lib 配下のフォルダーは次のとおりです。

- ia32: IA-32 上で動作するアプリケーションのビルドに使用するファイル
- intel64: インテル® 64 上で動作するアプリケーションのビルドに使用するファイル
- ia64: IA-64 上で動作するアプリケーションのビルドに使用するファイル

インテル® C++ コンパイラーとインテル® Fortran コンパイラーの両方がインストールされている場合、所定のバージョンのフォルダーが共有されます。

## 2.3 削除/アンインストール

製品の削除 (アンインストール) は、製品をインストールしたユーザー (root または非 root ユーザー) で実行してください。インストールに `sudo` を使用した場合は、アンインストールの際にも使用する必要があります。インストールされているパフォーマンス・ライブラリー・コンポーネントや Eclipse 統合コンポーネントを残したまま、コンパイラーのみを削除することはできません。

1. 端末を開いて、`<install-dir>` 以外のフォルダーに移動 (`cd`) します。
2. コマンド `<install-dir>/bin/ia32/uninstall_cproc.sh` を入力します (必要に応じて `ia32` を `intel64` または `ia64` に変更してください)。
3. 画面の指示に従ってオプションを選択します。
4. 別のコンポーネントを削除するには、ステップ 2 と 3 を繰り返します。

同じバージョンの Intel® Fortran コンパイラーをインストールしている場合は、Fortran コンパイラーも削除されます。使用している Eclipse に Intel® C++ コンパイラーの Eclipse 統合機能が追加されている場合は、Eclipse の構成から Intel の統合拡張を削除して、構成を更新する必要があります。

## 3 Intel® C++ コンパイラー

このセクションでは、Intel® C++ コンパイラーの変更点、新機能、および最新情報をまとめられています。

### 3.1 互換性

バージョン 11.0 では、IA-32 システムのデフォルトでのコード生成において、アプリケーションを実行するシステムで Intel® ストリーミング SIMD 拡張命令 2 (Intel® SSE2) がサポートされていると仮定するように変更されました。詳細は、[下記を参照](#)してください。

### 3.2 新機能と変更された機能

詳細は、コンパイラーのドキュメントを参照してください。

- C++0x からの追加機能
- C++ ラムダ関数
- 並列実行用の言語拡張
- 非同期 I/O 拡張
- 10 進浮動小数点
- `#pragma vector_nontemporal`
- `#pragma unroll_and_jam`
- IPP オプションを使用した `valarray` の実装

### 3.3 新規および変更されたコンパイラー・オプション

- `-diag-enable sc-include`
- `-diag-enable sc-parallel`
- `-mkl[=lib]`
- `-xAVX`

廃止予定のコンパイラー・オプションのリストは、ドキュメントのコンパイラー・オプションのセクションを参照してください。

### 3.3.1 -O0 オプションの -mp の除外

バージョン 11.1 では、最適化を無効にする -O0 オプションは、浮動小数点精度を最大にする -mp を含意しなくなりました。-mp スイッチは廃止予定です。そのため、浮動小数点精度の影響を受けやすいアプリケーションには、明示的に -fp-model オプションを指定することを推奨します。

## 3.4 その他の変更

### 3.4.1 最適化レポートがデフォルトで無効に設定

バージョン 11.1 以降、コンパイラーは、ベクトル化、自動並列化、OpenMP スレッド化ループに関する最適化レポートメッセージをデフォルトで表示しないようになりました。これらのメッセージを表示するには、-diag-enable vec、-diag-enable par、-diag-enable openmp を指定するか、-vec-report、-par-report、-openmp-report を使用する必要があります。

また、バージョン 11.1 以降、最適化レポートメッセージは stdout ではなく、stderr に送られます。

### 3.4.2 コンパイラー環境の構築

コマンドライン・ビルド環境の設定に使用されていた iccvars.sh (iccvars.csh) スクリプトが変更されました。以前のバージョンでは、cc または cce のいずれかのルート・ディレクトリーを選択することによってターゲット・プラットフォームが選択されました。バージョン 11.x では、スクリプトは 1 つのみで、引数を指定してターゲット・プラットフォームを選択します。

コマンドの形式は以下のとおりです。

```
source <install-dir>/Compiler/11.1/xxx/bin/iccvars.sh argument
```

<install-dir> はインストール・ディレクトリー (デフォルトのインストール先は /opt/intel) で、xxx はビルド番号です。argument は、ia32、intel64、ia64 のいずれかです (「[インストール先フォルダー](#)」を参照)。コンパイラー環境を構築すると、インテル® デバッガー (idb) 環境も構築されます。

### 3.4.3 デフォルトの命令セットがインテル® ストリーミング SIMD 拡張命令 2 (インテル® SSE2) を必要とするものに変更

バージョン 11.0 以降、IA-32 アーキテクチャー向けのコンパイルでは、-msse2 (旧: -xW) がデフォルトになりました。-msse2 でビルドされたプログラムは、インテル® Pentium® 4 プロセッサや特定のインテル以外のプロセッサなど、インテル® ストリーミング SIMD 拡張命令 2 (インテル® SSE2) をサポートするプロセッサ上で実行する必要があります。互換性を保証するランタイムチェックは行われません。プログラムがサポートされていないプロセッサで実行されている場合は、無効な命令フォルトが発生する場合があります。これにより、インテル® SSE 命令が x87 命令の代わりに使用され、高い精度ではなく、宣言された精度で計算が行われることがあるため、浮動小数点結果が変更される可能性があることに注意してください。

すべてのインテル® 64 アーキテクチャー・プロセッサでインテル® SSE2 がサポートされています。

汎用 IA-32 の以前のデフォルトを使用する場合は、-mia32 を指定してください。

### 3.4.4 OpenMP ライブラリーのデフォルトが "compat" に変更

バージョン 10.1 では、新しい OpenMP ライブラリー・セットが追加され、アプリケーションは、インテル® コンパイラーと Microsoft\* コンパイラーの両方からの OpenMP コードを使用することが可能でした。この "互換" ライブラリーは古い "レガシー" ライブラリーよりも高いパフォーマンスを提供します。バージョン 11.x では、互換ライブラリーが OpenMP アプリケーションのデフォルト・ライブラリーとして使用されるようになりました。

-openmp-lib compat と等価です。古いライブラリーを使用する場合は、-openmp-lib legacy を指定してください。

"レガシー" ライブラリーは、インテル® コンパイラーの将来のリリースからは削除される予定です。

### 3.4.5 mathf.h の削除

以前は、ヘッダーファイルの mathf.h が単精度マス・ライブラリー関数の定義に使用されていましたが、本製品からは削除されています。このヘッダーファイルを使用していた場合は、代わりに mathimf.h を使用してください。

### 3.4.6 サンプリング・ベースのプロファイルに基づく最適化機能の削除

ハードウェア・サンプリング・ベースのプロファイルに基づく最適化機能は提供されなくなりました。この変更に伴い、-prof-gen-sampling と -ssp の 2 つのコンパイラー・オプション、および profrun と pronto\_tool の 2 つの実行ファイルが削除されました。インストルメント形式のプロファイルに基づく最適化機能は従来どおり利用できます。

### 3.4.7 OpenMP ヘッダーファイルの変更

バージョン 11.1 Update 4 では、OpenMP ヘッダーファイルの omp.h にエラーチェックが追加されました。

omp\_lock\_t 型と omp\_nest\_lock\_t 型の定義が変更されています。本リリースでは、コンパイラーはこれらの型をコンパイル時に識別します。この変更は、OpenMP 仕様に準拠した OpenMP プログラムには影響しません。OpenMP 仕様に準拠していない OpenMP プログラムでは、コンパイラーが警告を発行することがあります。次に例を示します。

```
$ cat sample.c
#include <omp.h>
int main() {
    omp_lock_t lk;
    omp_init_nest_lock( &lk );
    return 0;
} // main

$ icc -openmp sample.c

sample.c(4): warning #167: argument of type "omp_lock_t *" is
incompatible with parameter of type "omp_nest_lock_t *"

    omp_init_nest_lock( &lk );
                        ^
```

## 3.5 Eclipse IDE でのソースチェッカーの使用

IDE 内でソースチェッカー (旧: スタティックの検証) サポートを有効にすると、最終的なビルドターゲット (例: 実行ファイル) は作成されません。スタティックの検証が必要な場合は、デバッグ (開発) 構成のコピーを作成して、「ソースチェッカー」の構成を別途作成することを推奨します。

- プロジェクトのプロパティ・ページを開いて、[C/C++ Build (C/C++ ビルド)] を選択します。
- [Manage… (管理)] ボタンをクリックします。
- [Manage (管理)] ダイアログで [New… (新規)] ボタンをクリックして、[Create configuration (構成を作成)] ダイアログを開きます。
- [Name (名前)] ボックスに新しい構成の名前を入力します。
- 必要に応じて、[Description (記述)] に構成の説明を入力します。
- [Copy settings from (設定のコピー元)]、[Default configuration (デフォルト構成)]、または [Existing configuration (既存構成)] ラジオボタンをクリックして、対応するドロップダウンメニューから設定を選択します。
- [O.K.] をクリックして [Create configuration (構成を作成)] ダイアログを閉じます。
- [O.K.] をクリックして (新しい設定名が選択されている) [Manage (管理)] ダイアログを閉じます。
- プロパティ・ページに新しい設定が表示されます。また、新しい設定がアクティブなビルドの設定に指定されます。
- コンパイラの [Compilation Diagnostics (コンパイル診断)] を表示します。ソースコードの解析を制御するには、[Level of Source Code Analysis (ソースコード解析のレベル)]、[Level of Source Code Parallelization Analysis (ソースコード並列化解析のレベル)]、[Analyze Included Files (インクルード・ファイルの解析)] プロパティを使用します。

## 3.6 既知の問題

### 3.6.1 TR1 システムヘッダー

g++ バージョン 4.3 以降がインストールされているシステムで TR1 (C++ Library Technical Report 1) システムヘッダーを使用している場合、インテル® C/C++ コンパイラーは、<type\_traits> ヘッダーファイルのコンパイルの際にエラーを出力します。これは、インテル® C/C++ コンパイラーでは、可変個引数テンプレートと呼ばれる C++0x 機能をサポートしていないためです。次のようなコンパイルエラーが表示されます。

```
../include/c++/4.3.0/tr1_impl/type_traits(170): エラー: 識別子を指定してください。
```

```
template<typename _Res, typename... _ArgTypes>
```

^

```
include/c++/4.3.0/tr1_impl/type_traits(171): エラー: ")" を指定してください。
```

```
struct __is_function_helper<_Res(_ArgTypes...)>
```

これらのヘッダーを使用しないか、または古いバージョンの g++ コンパイラーを使用してください。

### 3.6.2 KMP\_AFFINITY のデフォルト動作の変更

スレッド・アフィニティー型の KMP\_AFFINITY 環境変数のデフォルトは none (KMP\_AFFINITY=none) です。KMP\_AFFINITY=none の動作は、10.1.015 以降で変更されており、すべての 11.x コンパイラーでは、初期化スレッドによりマシン上の全スレッドの「フルマスク」が作成され、起動時に各スレッドはこのマスクにバインドします。この変更により、その他のプラットフォームのアフィニティー・メカニズム (SGI Altix マシンの `dplace()` など) に影響する可能性があることが判明しました。この問題を解決するため、新しいアフィニティー型の `disabled` がコンパイラー 10.1.018 とすべての 11.1 コンパイラー (KMP\_AFFINITY=disabled) で導入されています。KMP\_AFFINITY=disabled を設定すると、OpenMP ランタイム・ライブラリーによるアフィニティー関連のシステムコールが回避されます。

### 3.6.3 古いバージョンの ld による致命的なエラー

状況により、アプリケーションをバージョン 11.x コンパイラーにリンクすると次のような内部エラーが発生し、アプリケーションが失敗することがあります。

```
ld: BFD 2.15.92.0.2 20040927 内部エラー。bfd_get_reloc_size
の ../../bfd/reloc.c 行 444 で終了します。
ld: この問題を報告してください。
```

この問題を解決するには、最新バージョンの binutils をインストールしてください。推奨する最小バージョンは、2.17.50 です。

### 3.6.4 一部の Intel® AVX アーキテクチャー命令と組み込み命令の削除

VPERMIL2PD 命令と VPERMIL2PS 命令はサポートされていません。また、対応する組み込み命令 `_mm256_permute2_pd`、`_mm_permute2_pd`、`_mm256_permute2_ps`、`_mm_permute2_ps` もサポートされていません。これらの命令や組み込み命令は Intel® AVX アーキテクチャーから削除されていますが、コンパイラー・ドキュメントからはその記述が削除されていません。詳細は、<http://software.intel.com/en-us/blogs/2009/01/29/recent-intelr-avx-architectural-changes/> (英語) を参照してください。

## 4 Intel® デバッガー (IDB)

次の注意事項は、IA-32 アーキテクチャー・システムおよび Intel® 64 アーキテクチャー・システムで実行する Intel® デバッガー (IDB) のグラフィカル・ユーザー・インターフェイス (GUI) についてです。このバージョンでは、`idb` コマンドは GUI を起動します。コマンドライン・インターフェイスを起動するには、`idbc` を使用します。

IA-64 アーキテクチャー・システムでは、GUI は利用できません。`idb` コマンドはコマンドライン・インターフェイスを起動します。

### 4.1 Java ランタイム環境の設定

Intel® IDB デバッガーのグラフィカル環境は、Java アプリケーションで構築されており、実行には Java ランタイム環境 (JRE) が必要です。デバッガーは、5.0 (1.5) または 6.0 (1.6) JRE をサポートしています。

配布元の手順に従って JRE をインストールします。

最後に、JRE のパスを設定する必要があります。

```
export PATH=<path_to_JRE_bin_dir>:$PATH
```

## 4.2 デバッガーの起動

デバッガーを起動するには、まず始めに、「[コンパイラー環境の構築](#)」で説明されているコンパイラー環境が構築されていることを確認してください。その後、次のコマンドを使用します。

```
idb
```

または

```
idbc
```

(必要に応じて)

GUI が開始され、コンソールウィンドウが表示されたら、デバッグセッションを開始できます。

**注:** デバッグする実行ファイルが、デバッグ情報付きでビルドされ、実行可能ファイルであることを確認してください。必要に応じて、アクセス権を変更します。

例:

```
chmod +x <application_bin_file>
```

## 4.3 その他のドキュメント

インテル® コンパイラー / インテル® デバッガー・オンライン・ヘルプは、デバッガーのグラフィカル・ユーザー・インターフェイスの [Help (ヘルプ)] > [Help Contents (ヘルプ目次)] で表示できます。

[Help (ヘルプ)] ボタンが表示されているデバッガーのダイアログから状況依存ヘルプにもアクセスできます。

## 4.4 デバッガー機能

### 4.4.1 IDB の主な機能

デバッガーは、インテル® IDB デバッガーのコマンドライン・バージョンのすべての機能をサポートしています。デバッガー機能は、デバッガー GUI または GUI コマンドラインから呼び出すことができます。グラフィカル環境を使用する場合は、既知の制限を参照してください。

### 4.4.2 新機能と変更された機能

- IA-32 およびインテル® 64 アーキテクチャー用のデバッガー GUI
- 並列実行デバッグサポート
- セッションコンセプト
- ビットフィールド・エディター
- SIMD レジスターウィンドウ
- OpenMP サポート
  - タスク、バリア、タスクウェイト、ロック、チームおよびタスク・スポン・ツリーの情報ウィンドウ
  - データ共有イベントと再入可能な呼び出しの検出
  - 再コンパイル不要のシリアルコードのデバッグ
- 国際化サポート

## 4.5 既知の問題

### 4.5.1 データ共有検出の問題

- [Stop on Event (イベントで停止)] アイコンが無効な場合、または [Parallel (並列)] > [Stop on Event (イベントで停止)] メニュー項目がオフの場合、データ共有イベントは [Data Sharing Events (データ共有イベント)] ウィンドウでは収集されません。デバッガーを停止し、[Data Sharing Events (データ共有イベント)] ウィンドウを開いた場合、最後のイベントのみが表示されます。
- [Data Sharing Events (データ共有イベント)] ウィンドウを閉じ、再度開くと、最近のイベントと重複する新しい解析実行ノードが表示されます。
- データ共有検出中に [Data Sharing Events (データ共有イベント)] ウィンドウを閉じると、検出後、ウィンドウを再度開いたときに最後のイベントのみが表示されます。

### 4.5.2 [Signals (シグナル)] ダイアログが動作しない

GUI ダイアログの [Debug (デバッグ)] > [Signal Handling (シグナル処理)], またはショートカット・キーの Ctrl+S でアクセス可能な [Signals (シグナル)] ダイアログが正しく動作しないことがあります。シグナル・コマンドライン・コマンドを代わりに使用する場合は、インテル® デバッガー (IDB) マニュアルを参照してください。

### 4.5.3 再入可能呼び出しにおけるブレイクの未サポート

このバージョンのデバッガーでは、再入可能呼び出しでブレイク機能はサポートされていません。[再入可能呼び出しでブレイク...] メニューアイコンをクリックするか、メニューから [Parallel (並列)] > [Break on Re-entrant Call... (再入可能呼び出しでブレイク...)] を選択して、条件を設定することはできますが、再入可能呼び出しが発生してもデバッガーは結果を表示しません。

### 4.5.4 GUI のサイズ調整

デバッガーの GUI ウィンドウのサイズが小さくなり、一部のウィンドウが表示されていないことがあります。ウィンドウを拡大すると、隠れているウィンドウが表示されます。

### 4.5.5 プロセスの終了

デバッガーの実行中は、[Debug (デバッグ)] メニューの [Kill Focused Process (フォーカスがあるプロセスの終了)] コマンドは動作しません。最初にデバッガーを停止してから、プロセスを終了してください。

### 4.5.6 10 進浮動小数点の未サポート

インテル® デバッガーでは、一部の C++ コンパイラーでサポートされている 10 進浮動小数点データ型はサポートされていません。そのため、デバッガーではそのような変数は文字配列のように表示されます。

### 4.5.7 \$cdir ディレクトリー、\$cwd ディレクトリー

\$cdir はコンパイル・ディレクトリーです (記録されている場合)。\$cdir は、ディレクトリーが設定されている場合にサポートされます。シンボルとしてサポートされるわけではありません。

\$cwd は現在の作業ディレクトリーです。セマンティクスもシンボルもサポートされていません。

\$cwd と '.' の違いは、\$cwd はデバッグセッション中に変更された現在の作業ディレクトリーを追跡する点です。 '.' は、ソースパスへのエントリーが追加されると直ちに現在のディレクトリーに展開されます。

#### 4.5.8 info stack の使用

デバッガーコマンド `info stack` は、以下のオプションの構文では現在、負のフレームカウントをサポートしていません。

```
info stack [num]
```

フレームカウント `num` が正の場合、最内 `num` フレームを出力します。カウントが負またはゼロの場合、(最外 `num` フレームを出力するのではなく) フレームを出力しません。

#### 4.5.9 \$stepg0 のデフォルト値の変更

デバッガー変数 `$stepg0` のデフォルト値が 0 に変更されました。値 "0" の設定では、"step" コマンドを使用する場合、デバッガーはデバッグ情報なしでコードにステップオーバーします。以前のデバッガーバージョンと互換性を保つようするには、次のようにデバッガー変数を 1 に設定します。

```
(idb) set $stepg0 = 1
```

#### 4.5.10 一部の Linux システムでの SIGTRAP エラー

特殊な Linux カーネルを使用する一部のケースでは、デバッガーがブレークポイントで停止した後、ユーザーがデバッグを続行すると SIGTRAP エラーが発生することがあります。この問題を回避するには、SIGTRAP シグナルを次のようにコマンドラインで定義します。

```
(idb) handle SIGTRAP nopass noprint nostop
SIGTRAP is used by the debugger.
SIGTRAP      No      No      No      Trace/breakpoint
trap
(idb)
```

#### 4.5.11 MPI プロセスのデバッグには idb GUI は使用できません

MPI プロセスのデバッグに `idb GUI` を使用することはできません。コマンドライン・インターフェイス (`idbc`) を使用してください。

## 5 Eclipse 統合

IA-32 およびインテル® 64 アーキテクチャー向けインテル® C++ コンパイラーでは、Eclipse 機能と関連プラグイン (インテル® C++ Eclipse 製品拡張) がインストールされます。これらを Eclipse 統合開発環境 (IDE) として追加すると、インテル® C++ コンパイラーが Eclipse でサポートされます。これにより、インテル® C++ コンパイラーを Eclipse 統合開発環境から使用して、アプリケーションを開発することができます。

## 5.1 提供されている統合

### 5.1.1 Eclipse 3.5 および CDT 6.0

Eclipse プラットフォームのバージョン 3.3.x 用のファイルは次のディレクトリーにあります。

```
<install-dir>/eclipse_support/cdt6.0/eclipse
```

統合には、Eclipse プラットフォームのバージョン 3.5.x、Eclipse C/C++ Development Tools (CDT) のバージョン 6.0.0 以降、および Java ランタイム環境 (JRE) (5.0 (1.5) 以降、6.0 を推奨) が必要です。

### 5.1.2 Eclipse 3.4 および CDT 5.0

Eclipse プラットフォームのバージョン 3.3.x 用のファイルは次のディレクトリーにあります。

```
<install-dir>/eclipse_support/cdt5.0/eclipse
```

統合には、Eclipse プラットフォームのバージョン 3.4.x、Eclipse C/C++ Development Tools (CDT) のバージョン 5.0.0 以降、および Java ランタイム環境 (JRE) (5.0 (1.5) 以降、6.0 を推奨) が必要です。

### 5.1.3 統合に関する注意事項

Eclipse プラットフォームのバージョン 3.5 と 3.4 は、現在 IA-64 アーキテクチャーでは利用できないことに注意してください。このアーキテクチャー用の Eclipse 統合が含まれているコンパイラー・キットは、今後リリースされるプラットフォームのものであります。

すでに適切なバージョンの Eclipse、CDT、および JRE が環境にインストールされ、設定されている場合は、このセクションの「[Eclipse でのインテル® C++ Eclipse 製品拡張のインストール方法](#)」で説明するように、インテル® C++ Eclipse 製品拡張を Eclipse に追加インストールできます。そうでない場合は、このセクションの「[Eclipse、CDT、および JRE の入手方法とインストール方法](#)」で説明するように、最初に Eclipse、CDT、および JRE を入手して、インストールしてください。そして、その後インテル® C++ Eclipse 製品拡張をインストールします。

## 5.2 Eclipse でのインテル® C++ Eclipse 製品拡張のインストール方法

既存の Eclipse の構成にインテル® C++ Eclipse 製品拡張を追加するには、Eclipse から次の手順を実行します。

### 5.2.1 Eclipse 3.5.0 および CDT 6.0.0 “Galileo”

[Available Software (利用可能なソフトウェア)] ページ ([Help (ヘルプ)] > [Install New Software... (新規ソフトウェアのインストール...)]) を開きます。[Add... (追加...)] ボタンをクリックして、[Local... (ローカル...)] を選択します。ディレクトリー・ブラウザが開きます。インテル® C++ コンパイラーのインストール・ディレクトリーにある eclipse ディレクトリーを選択します。例えば、root としてコンパイラーをデフォルトのディレクトリーにインストールした場合は、`/opt/intel/Compiler/11.1/uuu/eclipse_support/cdt6.0/eclipse` を選択します。[OK] をクリックして、ディレクトリー・ブラウザを閉じます。[OK] をクリックして、[Add Site (サイトの追加)] ダイアログを閉じ、インテル® C++ 統合機能の 2 つのボックスを選択します。1 つめは [Intel® C++ Compiler Documentation (インテル® C++ コンパイラー・ドキュメント)]、2 つめは [Intel® C++ Compiler Professional 11.1 for Linux\* (インテル® C++ コンパイラー 11.1 Linux\* 版)] です。

**注:** [Group items by category (項目をカテゴリー別にグループ化)] が設定されている場合には、インテルの機能は表示されません。インテルの機能を表示するには、このオプションをオフにしてください。

[Install (インストール)] ボタンをクリックします。[Install (インストール)] ダイアログが表示され、インストールする項目を確認できます。[Next (次へ)] をクリックします。契約に同意するかどうかを確認するメッセージが表示されます。契約に同意したら、[Finish (完了)] をクリックします。インストールが開始します。

Eclipse の再起動を求められたら、[Yes (はい)] を選択します。Eclipse が再起動したら、インテル® C++ コンパイラーを使用する CDT プロジェクトを作成して作業することができます。詳細は、インテル® C++ コンパイラーのドキュメントを参照してください。インテル® C++ コンパイラーのドキュメントは、[Help (ヘルプ)] > [Help Contents (ヘルプ目次)] > [Intel C++ Compiler User Guide (インテル® C++ コンパイラー・ユーザー・ガイド)] から表示できます。インテル® デバッガー (idb) とともに idb Eclipse 製品拡張をインストールし、Eclipse 内で idb を使用する場合は、同じ方法で idb 製品拡張サイトを Eclipse に追加します。例えば、root としてキットをデフォルトのディレクトリーにインストールした場合、idb Eclipse 製品拡張は /opt/intel/Compiler/11.1/uuu/idb/eclipse\_support/cdt6.0/eclipse にインストールされます。

## 5.2.2 Eclipse 3.4.0 および CDT 5.0.0 "Ganymede"

次のメニューを選択して、[Software Updates and Add-ons (ソフトウェア更新とアドオン)] ページを開きます。

[Help (ヘルプ)] > [Software Updates... (ソフトウェア更新)]

[Available Software (利用可能なソフトウェア)] タブを開きます。

[Add Site... (サイトの追加)] > [Local... (ローカル)] を選択します。ディレクトリー・ブラウザが開きます。インテル® C++ コンパイラーのインストール・ディレクトリーにある eclipse ディレクトリーを選択します。例えば、root としてコンパイラーをデフォルトのディレクトリーにインストールした場合は、/opt/intel/Compiler/11.1/uuu/eclipse\_support/cdt5.0/eclipse を選択します。

[OK] をクリックして、ディレクトリー・ブラウザを閉じます。[OK] をクリックして、[Add Site (サイトの追加)] ダイアログを閉じ、インテル® C++ 統合機能の 2 つのボックスを選択します。1 つめは [Intel® C++ Compiler Documentation (インテル® C++ コンパイラー・ドキュメント)]、2 つめは [Intel® C++ Compiler 11.1 for Linux\* (インテル® C++ コンパイラー 11.1 Linux\* 版)] です。

**注:** [Group items by category (項目をカテゴリー別にグループ化)] が設定されている場合には、インテルの機能は表示されません。インテルの機能を表示するには、このオプションをオフにしてください。

[Install (インストール)] ボタンをクリックします。[Install (インストール)] ダイアログが表示され、インストールする項目を確認できます。[Next (次へ)] をクリックします。契約に同意するかどうかを確認するメッセージが表示されます。契約に同意したら、[Finish (完了)] をクリックします。インストールが開始します。

Eclipse の再起動を求められたら、[Yes (はい)] を選択します。Eclipse が再起動したら、インテル® C++ コンパイラーを使用する CDT プロジェクトを作成して作業することができます。詳細は、インテル® C++ コンパイラーのドキュメントを参照してください。インテル® C++ コンパイラーのドキュメントは、[Help (ヘルプ)] > [Help Contents (ヘルプ目次)] > [Intel C++ Compiler User and Reference Guides (インテル® C++ コンパイラー・ユーザー・リファレンス・ガイド)] で表示できます。

インテル® デバッガー (idb) とともに idb Eclipse 製品拡張をインストールし、Eclipse 内で idb を使用する場合は、同じ方法で idb 製品拡張サイトを Eclipse に追加します。例えば、root としてキットをデフォルトのディレクトリーにインストールした場合、idb Eclipse 製品拡張は /opt/intel/Compiler/11.1/uuu/idb/eclipse\_support/cdt5.0/eclipse にインストールされます。

## 5.3 Eclipse、CDT、および JRE の入手方法とインストール方法

Eclipse は Java アプリケーションのため、実行には Java ランタイム環境 (JRE) が必要です。Eclipse 3.4.0 の実行には JRE 5.0 (1.5) が必要です。インテルでは、JRE 6.0 (1.6) を使用することを推奨します。JRE は、オペレーティング環境 (マシン・アーキテクチャー、オペレーティング・システムなど) に応じてバージョンを選択します。また、多くの JRE の中から選択可能です。

### 5.3.1 Eclipse 3.5.0 および CDT 6.0.0

Eclipse 3.5.0 および CDT 6.0.0 の両方が含まれたパッケージは、以下の Web サイトから入手できます。

<http://www.eclipse.org/downloads/packages/eclipse-ide-cc-developers/galileor> (英語)

必要に応じて、Linux 32bit または Linux 64bit をダウンロードしてください。

### 5.3.2 Eclipse 3.4.0 および CDT 5.0.0

Eclipse 3.4.0 および CDT 5.0.0 の両方が含まれたパッケージは、以下の Web サイトから入手できます。

<http://www.eclipse.org/downloads/packages/release/ganymede/r> (英語)

「Eclipse IDE for C/C++ Developers」から、必要に応じて Linux 32bit または Linux 64bit をダウンロードしてください。

### 5.3.3 JRE、Eclipse、CDT のインストール

適切なバージョンの Eclipse、CDT、および JRE をダウンロードしたら、次の手順に従ってインストールします。

1. 配布元の手順に従って、JRE をインストールします。
2. Eclipse をインストールするディレクトリーを作成し、cd でこのディレクトリーに移動します。ここでは、このディレクトリーを <eclipse-install-dir> と表記します。
3. Eclipse パッケージのバイナリー、.tgz ファイルを <eclipse-install-dir> ディレクトリーにコピーします。
4. .tgz ファイルを展開します。
5. eclipse を起動します。

これで、Eclipse の構成にインテル® C++ 製品拡張を追加する準備が完了です。追加する方法は、「Eclipse でのインテル® C++ Eclipse 製品拡張のインストール方法」のセクションで説明されています。Eclipse の初回起動時のヘルプが必要な場合は、次のセクションを参照してください。

## 5.4 インテル® C++ コンパイラーで開発するための Eclipse の起動

LANG 環境変数を設定していない場合は、設定してください。次に例を示します。

```
setenv LANG ja_JP.UTF8
```

Eclipse を起動する前に `iccvvars.csh` (または `.sh`) スクリプトを実行して、インテル® C++ コンパイラー関連の環境変数を設定します。

```
source <install-dir>/bin/iccvvars.csh arch_arg ("arch_arg" は "ia32" または "intel64" のいずれか)
```

Eclipse を実行するには JRE が必要なため、Eclipse を起動する前に JRE が利用可能であることを確認してください。PATH 環境変数の値をシステムにインストールされている JRE の `java` ファイルのフォルダーへのフルパスに設定するか、Eclipse コマンドの `-vm` パラメーターでシステムにインストールされている JRE の `java` 実行ファイルへのフルパスを参照します。

例:

```
eclipse -vm /JRE folder/bin/java
```

Eclipse がインストールされているディレクトリーから Eclipse 実行ファイルを直接起動します。次に例を示します。

```
<eclipse-install-dir>/eclipse/eclipse
```

## 5.5 Fedora システムでのインストール

root アカウントではなくローカルアカウントとして、インテル® C++ コンパイラー Linux 版を Fedora 搭載の IA-32 またはインテル® 64 システムにインストールすると、Eclipse を起動する際に、コンパイラーまたはデバッガーで Eclipse グラフィカル・ユーザー・インターフェイスが正しく表示されないことがあります。この場合、通常、JVM Terminated エラーが表示されます。また、システムレベルの root アカウントでソフトウェアをインストールし、それ以下の権限のユーザーアカウントで実行する場合もエラーが発生します。

これは、Fedora に実装されているセキュリティーのレベルが低いためです。この新しいセキュリティーは、ダイナミック・ライブラリーなど、システムリソースへのアクセスに悪影響を及ぼすことがあります。一般ユーザーがコンパイラーを使用するためには、システム管理者は SELinux セキュリティーを調整する必要があります。

## 5.6 コンパイラー・バージョンの選択

Eclipse プロジェクトでは、異なるバージョンのインテル® C++ コンパイラーがインストールされている場合、コンパイラーのバージョンを選択できます。IA-32 アーキテクチャー・システムでサポートされているインテル® コンパイラーのバージョンは、9.1、10.0、10.1、11.0、11.1 です。インテル® 64 アーキテクチャー・システムでは、コンパイラー・バージョン 11.0 と 11.1 がサポートされています。

## 6 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

このセクションでは、インテル® C++ コンパイラー・プロフェッショナル・エディションに同梱されているインテル® 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) の変更点、新機能、および最新情報をまとめています。インテル® IPP についての詳細は、次のリンクを参照してください。

- **新機能:** インテル® IPP 製品ページ (<http://software.intel.com/en-us/intel-ipp/> (英語)) を参照してください。
- **ドキュメント、ヘルプ、サンプル:** インテル® IPP 製品ページ (<http://software.intel.com/en-us/intel-ipp/>) のドキュメントのリンクを参照してください。

### 6.1 新機能と変更された機能

#### 6.1.1 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 6.1 Update 5

- 本リリースでは、新機能は追加されていません。報告されている問題が修正されています。

#### 6.1.2 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 6.1 Update 4

- 『インテル® IPP 信号処理リファレンス・マニュアル』にストリング処理に関する新しいコード例を追加
- ライブラリーに RSA-1024 暗号化の最適化を追加
- OpenSSL パフォーマンスの向上と OpenSSL 0.9.8j のサポート

#### 6.1.3 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 6.1 Update 3

- 『インテル® IPP 信号処理リファレンス・マニュアル』の第 11 章に新しいコード例を追加
- UMC ドキュメントに動き検出とモード決定のセクションを追加
- BZIP2 デコーダーでパフォーマンスが約 5% 向上

#### 6.1.4 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 6.1 Update 2

- インテル® Advanced Vector Extensions (インテル® AVX) のサポート
- インテル® Core™ i7 プロセッサの新しい最適化とスレッド化制御/最適化をサポート
- 3D 画像処理: 3D 幾何学変換、3D フィルター
- 新しいデータ圧縮関数 API
- RSA\_SSA1.5 と RSA\_PKCSv1.5 の新しいインテル® IPP 暗号化サポート
- PNG 形式サポートを追加する UIC (Unified Image Classes) と DXT1、DXT3、DXT5 画像圧縮をサポートする新しい機能
- 球面調和関数とパーリンノイズ生成関数を含む高度な光関数
- Windows Media\* Photo (HD Photo) のサポート: IPP PCT 関数
- MPEG-2 のシーン解析、VC1 の輝度補償とオーバーラップ・スムージングを含む新しいビデオ・コーディング分野の向上
- 信号処理、画像処理、ストリング処理、C++/C# 言語サポートのサンプルを /Samples フォルダに追加。その他のサンプルは、<http://software.intel.com/en-us/articles/intel-integrated-performance-primitives-intel-ipp-intel-ipp-sample-code/> (英語) からダウンロードできます。
- 廃止予定の API のさらに多くのリファレンス情報がリファレンス・マニュアルとヘッダーファイルに追加

## 6.2 既知の制限事項

- 問題の修正リスト、既知の問題、制限事項については、次のナレッジベースの記事を参照してください。  
<http://software.intel.com/en-us/articles/intel-ipp-library-61-fixes-list/> (英語)

## 6.3 別途ダウンロード可能なインテル® IPP 暗号化ライブラリー

インテル® IPP 暗号化ライブラリーは別途ダウンロード可能です。ダウンロードとインストールの手順については、次の Web サイトを参照してください。  
<http://software.intel.com/en-us/articles/download-ipp-cryptography-libraries/> (英語)

## 6.4 インテル® IPP コードサンプル

インテル® IPP コードサンプルとして、Windows\* 版、Linux 版、Mac OS\* 版のダウンロード・パッケージが用意されています。以下の Web サイトから入手できます。  
<http://software.intel.com/en-us/articles/intel-integrated-performance-primitives-code-samples/> (英語)

サンプルには、オーディオ/ビデオコーデック、画像処理、メディア・プレーヤー・アプリケーション、C++/C#/Java からの呼び出し関数のソースコードが含まれています。サンプルのビルド方法についての説明は、各サンプルのインストール・パッケージの readme ファイルをご覧ください。

## 7 インテル® マス・カーネル・ライブラリー

このセクションでは、インテル® C++ コンパイラー・プロフェッショナル・エディションに同梱されているインテル® マス・カーネル・ライブラリー (インテル® MKL) の変更点、新機能、および最新情報をまとめています。

### 7.1 本バージョンでの変更

本バージョンおよび以前のバージョンの変更についての詳細は、  
<http://software.intel.com/en-us/articles/new-in-intel-mkl-10-2/> (英語) を参照してください。

#### 7.1.1 インテル® マス・カーネル・ライブラリー 10.2 Update 5

##### 新機能

- LAPACK 3.2.1 アップデート (主に LAPACK 3.2 に関する修正) に対応

##### パフォーマンスの向上

- FFT
  - インテル® 64 アーキテクチャー上で 3 次元以上の複素数 FFT のパフォーマンスが向上
- VSL
  - 45nm インテル® Core™2 Duo プロセッサー以降の 64 ビット・ライブラリーで、MT19937 と MT2203 基本乱数ジェネレーター (BRNG) のパフォーマンスが向上

##### ユーザービリティとインターフェイスの向上

- uBLAS の例で Boost 1.41.0 のサポートを追加
- 対角優位ソルバー関数 (?DTSVB、?DTTRFB、?DTTRSB) に Fortran 95 インターフェイスを追加
- インプレース多次元クラスター FFT のメモリー消費を大幅に削減

インテル® C++ コンパイラー 11.1 Linux\* 版プロフェッショナル・エディション  
インストール・ガイドおよびリリースノート

## 7.1.2 インテル® マス・カーネル・ライブラリー 10.2 Update 4

### 新機能

- 単精度複素数の絶対値を求める SCABS1 関数を追加
- 部分的なピボット演算を使用する一般的なソルバー (?GTSV) よりも 2 倍高速な対角優位の三重対角方程式用の ?DTSVB ソルバーを追加
- 対角優位の三重対角方程式用の因数分解ルーチン (?DTTRFB) と 前方/後方代入ルーチン (?DTTRSB) を追加

### パフォーマンスの向上

- FFT
  - 8 または 13 の倍数の変換のパフォーマンスが向上
  - ベクトル長が 2 のべき乗でない 1D 複素数クラスター FFT を最適化
- VSL
  - デシメーションを必要とする畳み込み/相関演算のパフォーマンスが大幅に向上 (再リンクが必要。「[既知の問題](#)」を参照)

## 7.1.3 インテル® マス・カーネル・ライブラリー 10.2 Update 3

### パフォーマンスの向上

- BLAS
  - インテル® Core™ i7 プロセッサおよびインテル® Xeon® プロセッサ 5300 番台、5400 番台、5500 番台で次の BLAS レベル 1、2 関数の 32 ビット OS バージョンがスレッド化: (D,S,C,Z)COPY、(D,S,C,Z)SWAP、(D,S,C,Z)AXPY、(S,C)ROT、(S,C)DOT、CDOTC、(D,S,C,Z)GEMV、(D,S,C,Z)TRMV、(S,C)SYMV、(S,C)SYR、(S,C)SYR2
  - インテル® Xeon® プロセッサ 5300、5400、5500 で次の BLAS レベル 1 関数の 32 ビットおよび 64 ビット OS バージョンが向上: ZAXPY、ZSCAL、ZDOT(U,C)、(D,S)ROT
  - インテル® Xeon® プロセッサ 5300 で列より非常に多い行を持つ行列の DGEMM スレッド化効率向上
- LAPACK
  - 次の LAPACK 関数のスケーラビリティ向上: ?POTRF、?GEBRD、?SYTRD、?HETRD、?STEDC 分割統治固有ソルバー
- FFT
  - 下層のカーネルが更新され、FFT において幅広くパフォーマンスが向上
  - 1 つの関数呼び出しで少数の変換が計算される場合に 3D FFT のスレッド化が向上
  - スレッド化を小規模の多次元変換に拡張
- VML
  - インテル® Xeon® プロセッサ 5500 番台の VML 関数がさらに最適化: v(s,d)Asin、v(s,d)Acos、v(s,d)Ln、v(s,d)Log10、vsLog1p、v[s/d]Hypot
- VSL
  - viRngPoisson および viRngPoissonV 乱数ジェネレーターのパフォーマンスが向上

### ユーザービリティとインターフェイスの向上

- uBLAS、Java、FFTW3、LAPACK95、BLAS95 のサンプルプログラムの向上
- examples ディレクトリーとリファレンス・マニュアルのソースコードにある同一のサンプルは、リファレンス・マニュアルでは削除
- クラスター FFT の新しい 64 ビット整数 (ILP64) fftw\_mpi インターフェイス

## 7.1.4 インテル® マス・カーネル・ライブラリー 10.2 Update 2

### 新機能

- LAPACK 3.2
  - 238 個の新しい LAPACK 関数
  - 超精密反復法の改良
  - ハウスホルダー QR 因数分解の非負対角
  - 低プロファイル行列でのハイパフォーマンス QR とハウスホルダー反射
  - 高速で正確な新しいヤコビ法 SVD
  - 矩形フル圧縮形式のルーチン
  - ピボットコレスキー
  - 混合精度反復法の改良 (コレスキー)
  - より安定した DQDS アルゴリズム
- DZGEMM 拡張 BLAS 関数の実装 (<http://www.netlib.org/blas/blast-forum/> の説明を参照)。リファレンス・マニュアルの BLAS セクションの \*gemm 関数ファミリーの説明を参照してください。
- PARDISO で実数、複素数、単精度データをサポート

### ユーザービリティ/インターフェイスの向上

- スパース行列形式変換ルーチン:
  - CSR (3-配列バリエーション) ↔ CSC (3-配列バリエーション)
  - CSR (3-配列バリエーション) ↔ 対角形式
  - CSR (3-配列バリエーション) ↔ スカイライン
- Fortran95 BLAS と LAPACK のコンパイル・モジュール・ファイル (.mod) を追加
  - モジュールは、インテル® Fortran コンパイラーで事前にビルドされており、インクルード・ディレクトリーにあります (フルパス情報については、インテル® MKL ユーザーズ・ガイドを参照してください)。
  - ほかのコンパイラー用のソースも提供されています。
  - インターフェイスについてのドキュメントは、インテル® MKL ユーザーズ・ガイドを参照してください。
- FFTW3 インターフェイスを直接メイン・ライブラリーに統合
  - デフォルトのインテル® Fortran コンパイラー規則と名前修飾で互換性のないコンパイラーでラッパーを作成するためのソースコードも提供されています。
  - 詳細は、リファレンス・マニュアルの付録 G を参照してください。
- DFTI\_DESCRIPTOR\_HANDLE を型の名前に追加。ユーザープログラムで型として参照できます。
- ユーザーデータへのアクセスを可能にするために最適化ソルバードメインのヤコビ行列計算ルーチンにパラメーターを追加 (詳細は、リファレンス・マニュアルの djacobix 関数の説明を参照してください)
- 64 ビット・アーキテクチャーでインテル® MKL の単精度 BLAS 関数 (頭文字 "s" または "c" の関数) から 64 ビット浮動小数点精度関数へのインターフェイス・マッピング呼び出しを追加 (詳細は、インテル® MKL ユーザーズ・ガイドの「sp2dp」を参照してください)
- 互換ライブラリー (「ダミーライブラリー」) を削除

## パフォーマンスの向上

- インテル® 64 アーキテクチャー用にさらにスレッド化された BLAS レベル 1、2 関数
  - レベル 1 関数 (ベクトル-ベクトル): (C,S,Z,D)ROT、(C,Z,S,D)COPY、(C,Z,S,D)SWAP
    - キャッシュのデータ位置に応じて、4 コアのインテル® Core™ i7 プロセッサ上でバージョン 10.1 Update 1 に対してパフォーマンスが最大 1.7 ~ 4.7 倍向上
    - キャッシュのデータ位置に応じて、24 コアのインテル® Xeon® プロセッサ 7400 番台システム上でバージョン 10.1 Update 1 に対してパフォーマンスが最大 14 ~ 130 倍向上
  - レベル 2 関数 (行列-ベクトル): (C,Z,S,D)TRMV、(S,D)SYMV、(S,D)SYR、(S,D)SYR2
    - キャッシュのデータ位置に応じて、4 コアのインテル® Core™ i7 プロセッサ上でバージョン 10.1 Update 1 に対してパフォーマンスが最大 1.9 ~ 2.9 倍向上
    - キャッシュのデータ位置に応じて、24 コアのインテル® Xeon® プロセッサ 7400 番台システム上でバージョン 10.1 Update 1 に対してパフォーマンスが最大 16 ~ 40 倍向上
- インテル® Core™ i7 プロセッサ、インテル® Xeon® プロセッサ (5300 番台、5400 番台、7400 番台) で、DSYRK の 32 ビット逐次バージョンに導入された再帰アルゴリズムのパフォーマンスが最大 20% 向上。
- インテル® Xeon® 7460 プロセッサで、大規模な問題の LU 因数分解 (DGETRF) がバージョン 10.1 Update 1 に対して 25% 向上。また小規模な問題でも劇的に向上。
- BLAS \*TBMV/\*TBSV 関数でレベル 1 BLAS 関数を使用。インテル® Core™ i7 プロセッサ上で最大 3%、インテル® Core™2 プロセッサ 5300 番台と 5400 番台で最大 10% のパフォーマンスが向上。
- DGEMM パフォーマンスを強化するスレッド化アルゴリズムの向上
  - 8 スレッドで最大 7% の向上、3、5、7 スレッドで最大 50% の向上 (インテル® Core™ i7 プロセッサ)
  - 3 スレッドで最大 50% の向上 (インテル® Xeon® プロセッサ 7400 番台)
- 非素数サイズのスレッド化 1D 複素数-複素数 FFT
- 3D 複素数-複素数変換の新しいアルゴリズムにより 1 スレッドまたは 2 スレッドで小さな問題サイズ (最大 64x64x64) についてより優れたパフォーマンスを提供
- 対称正定行列の演算時におけるアウトオブコア (OOC) PARDISO のハイレベルな並列化実装
- すべての行列の型でインコアとアウトオブコアの両方で PARDISO のメモリー使用量が減少
- 実対称行列、複素エルミート行列、複素対称行列に対し PARDISO OOC で使用されるメモリーがインテル® MKL 10.1 で使用されていたメモリーの半分以上まで減少
- PARDISO/DSS における順序付けの並列化とシンボリック因子分解
- インテル® Core® i7 プロセッサとインテル® Core™2 プロセッサで次の VML 関数において最大 2 倍のパフォーマンスの向上 (平均で 30% の向上): v(s,d)Round、v(s,d)Inv、v(s,d)Div、v(s,d)Sqrt、v(s,d)Exp、v(s,d)Ln、v(s,d)Atan、v(s,d)Atan2
- インテル® Advanced Vector Extension (インテル® AVX) で次の関数の最適化バージョンが利用可能
  - BLAS: DGEMM
  - FFT
  - VML: exp、log、pow
  - 上記の関数にアクセスする `mkl_enable_instructions()` 関数に関する重要な情報については、インテル® MKL ユーザーズ・ガイドを参照してください。

## 7.2 既知の問題

本リリースにおける既知の制限事項の詳細なリストは、<http://software.intel.com/en-us/articles/intel-math-kernel-library-support-resources/> (英語) を参照してください。

## 7.3 注意事項

インテル® MKL の将来のバージョンでは以下の変更が予定されています。「[テクニカルサポート](#)」を参照してください。

- ファイル名に `solver` を含むライブラリーの内容をコア・ライブラリーに移動する予定です。これらの `solver` ライブラリーはその後削除される予定です。

## 7.4 権利の帰属

エンド・ユーザー・ソフトウェア使用許諾契約書 (End User License Agreement) で言及されているように、製品のドキュメントおよび Web サイトの両方で完全なインテル製品名の表示 (例えば、“インテル® マス・カーネル・ライブラリー”) とインテル® MKL ホームページ ([www.intel.com/software/products/mkl](http://www.intel.com/software/products/mkl) (英語)) へのリンク/URL の提供を正確に行うことが最低限必要です。

インテル® MKL の一部の基となった BLAS の原版は <http://www.netlib.org/blas/index.html> (英語) から、LAPACK の原版は <http://www.netlib.org/lapack/index.html> (英語) から入手できます。LAPACK の開発は、E. Anderson、Z. Bai、C. Bischof、S. Blackford、J. Demmel、J. Dongarra、J. Du Croz、A. Greenbaum、S. Hammarling、A. McKenney、D. Sorensen らによって行われました。LAPACK 用 FORTRAN 90/95 インターフェイスは、<http://www.netlib.org/lapack95/index.html> (英語) にある LAPACK95 パッケージと類似しています。すべてのインターフェイスは、純粋なプロシージャー用に提供されています。

インテル® MKL クラスタ・エディションの一部の基となった ScaLAPACK の原版は <http://www.netlib.org/scalapack/index.html> (英語) から入手できます。ScaLAPACK の開発は、L. S. Blackford、J. Choi、A. Cleary、E. D’Azevedo、J. Demmel、I. Dhillon、J. Dongarra、S. Hammarling、G. Henry、A. Petitet、K. Stanley、D. Walker、R. C. Whaley らによって行われました。

インテル® MKL の PARDISO は、バーゼル大学 (University of Basel) から無償で提供されている PARDISO 3.2 (<http://www.pardiso-project.org> (英語)) と互換性があります。

本リリースのインテル® MKL の一部の FFT 関数は、カーネギーメロン大学からライセンスを受けて、SPIRAL ソフトウェア生成システム (<http://www.spiral.net/> (英語)) によって生成されました。本リリースのインテル® MKL の一部の FFT 関数は、ヒューストン大学からライセンスを受けて、UHFFT ソフトウェア生成システムによって生成されました。SPIRAL の開発は、Markus Püschel、José Moura、Jeremy Johnson、David Padua、Manuela Veloso、Bryan Singer、Jianxin Xiong、Franz Franchetti、Aca Gacic、Yevgen Voronenko、Kang Chen、Robert W. Johnson、Nick Rizzolo らによって行われました。

## 8 インテル® スレッディング・ビルディング・ブロック

このセクションでは、インテル® C++ コンパイラー・プロフェッショナル・エディションに同梱されているインテル® スレッディング・ビルディング・ブロック (インテル® TBB) 2.2 の変更点、新機能、および最新情報をまとめています。

## 8.1 本バージョンでの変更

### 8.1.1 インテル® スレッディング・ビルディング・ブロック 2.2 Update 2

- `parallel_invoke` と `parallel_for_each` で関数オブジェクトを値ではなく定数参照として利用可能
- 例外サポートの向上: 新しい例外クラスが追加され、すべての例外はアウトオブライン内部メソッドでスローされます。
- TBB のアロケータと `malloc` の置換に関する修正と向上 (安定したメモリー識別、Windows 上でのダイナミック関数置換を含む)
- `tbb_thread` クラスに `swap()` メソッドを追加
- `concurrent_hash_map` に `rehash()` メソッドと `bucket_count()` メソッドを追加
- コード、テスト、サンプル、ドキュメントでのその他の修正と追加/変更

### 8.1.2 インテル® スレッディング・ビルディング・ブロック 2.2 Update 1

- ドキュメントのアップデート
- TBB スケジューラーの自動初期化が可能性のあるすべての使用事例に対応
- `concurrent_queue`: パディングで使われる `sizeof` の引数の型が実際に使われる型と一致
- メモリー・アロケータの向上: `scalable_malloc` を呼び出す可能性のある `malloc` をサポート (Windows 以外)。TBB メモリー・アロケータ起動時のメモリー割り当て要求の処理を修正。
- Windows の `malloc` の置換でスタティック・オブジェクトのサポートが向上
- 並列処理できないパイプラインの起動時に、シングルスレッドでの実行を保証、アイドルスピンを排除、パフォーマンスが向上
- RML のリファクタリングとクリーンアップ
- 複数のアイテムの領域を予約するための `concurrent_hash_map` 用の新しいコンストラクターを追加
- TBB の例外クラスに `delete()` 演算子を追加
- `parallel_reduce` でのラムダサポートの強化
- `concurrent_queue` の gcc 4.3 警告の修正
- グローバル・スタティック・オブジェクトの構築中、TBB エンティティーを使用するモジュールで初期化時に発生する可能性のあるデッドロックを修正
- `concurrent_hash_map` のコピー・コンストラクターの修正
- 特定の場合にスケジューラーでクラッシュする問題の修正
- メモリー不足の場合に TBB アロケータでクラッシュする問題の修正
- ラムダ式がサポートされているかどうかをチェックする `test_lambda.cpp` を含む新しいテストの実装
- `concurrent_vector` の安全性問題に関する既知の例外の修正
- `concurrent_vector` における同時拡張要求の並列処理が向上
- TBB アロケータで大きなオブジェクトの割り当てのパフォーマンスがさらに向上
- Linux 上でのテキストの再配置ソースの問題を修正
- Windows 上での `malloc` の置換に関する問題を修正
- コード、テスト、サンプル、ドキュメントでのその他の小さな変更

## 8.2 既知の問題

インテル® スレッディング・ビルディング・ブロックの本リリースに関する次の注意事項に留意してください

- インテル® C++ コンパイラー 10.x を glibc 2.3.2、2.3.3、または 2.3.4 とともに使用したときに、TBB アルゴリズムまたはコンテナのコンテキストで実行されるユーザーコードで処理できない例外が発生すると、セグメンテーション違反が発生します。
- インテル® スレッド・チェッカーまたはインテル® スレッド・プロファイラーを使用した際により正確な結果を得るには、インテル® TBB とともに使用する前にそれらの製品の最新のアップデート・リリースをダウンロードしてください。
- 同じプログラムで連続してインテル® TBB と OpenMP コンストラクトをともに使用していて、OpenMP コードにインテル® コンパイラーを使用している場合、KMP\_BLOCKTIME に小さな値 (例えば、20 ミリ秒) を設定するとパフォーマンスが向上します。この設定は、`kmp_set_blocktime()` ライブラリー呼び出しを使用して OpenMP コード内で行うこともできます。KMP\_BLOCKTIME および `kmp_set_blocktime()` の詳細は、コンパイラーの OpenMP に関するドキュメントを参照してください。
- 一般に、アプリケーションやサンプルの非デバッグ ("リリース") ビルドは、インテル® TBB ライブラリーの非デバッグバージョンとリンクし、デバッグビルドはインテル® TBB ライブラリーのデバッグバージョンとリンクします。デバッグ・ライブラリーとリリース・ライブラリーの詳細については、製品ドキュメントのサブディレクトリーに含まれているチュートリアルを参照してください。
- Ubuntu 7.04 の 64 ビット・モードでコンパイルを行うと、エラーメッセージ "':system' has not been declared" が表示されます。この問題は、システムから `libpthread-dev` を削除することで回避できます。詳細は、<https://bugs.launchpad.net/ubuntu/+source/gcc-4.1/+bug/77559> (英語) を参照してください。

## 9 著作権と商標について

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。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を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、商適格性、あらゆる特許権、著作権、その他知的財産権の非侵害性への保証を含む)に関してもいかなる責任も負いません。インテルによる書面での合意がない限り、インテル製品は、その欠陥や故障によって人身事故が発生するようなアプリケーションでの使用を想定した設計は行われていません。

インテル製品は、予告なく仕様や説明が変更されることがあります。機能または命令の一覧で「留保」または「未定義」と記されているものがありますが、その「機能が存在しない」あるいは「性質が留保付である」という状態を設計の前提にしないでください。これらの項目は、インテルが将来のために留保しているものです。インテルが将来これらの項目を定義したことにより、衝突が生じたり互換性が失われたりしても、インテルは一切責任を負いません。この情報は予告なく変更されることがあります。この情報だけに基づいて設計を最終的なものとししないでください。

本書で説明されている製品には、エラッタと呼ばれる設計上の不具合が含まれている可能性があります。公表されている仕様とは異なる動作をする場合があります。現在確認済みのエラッタについては、インテルまでお問い合わせください。

最新の仕様をご希望の場合や製品をご注文の場合は、お近くのインテルの営業所または販売代理店にお問い合わせください。

本書で紹介されている注文番号付きのドキュメントや、インテルのその他の資料を入手するには、1-800-548-4725(アメリカ合衆国)までご連絡いただくか、インテルの Web サイトを参照してください。

本製品の一部は、オープンソースのライブラリーを使用してビルドされています。これらのライブラリーのライセンス規約に従い、インテルでは本製品のユーザーがライブラリーを利用できるようにしています。ライブラリーは、インテル® ソフトウェア開発製品のナレッジベース記事 (<http://software.intel.com/en-us/articles/open-source-downloads/>) からダウンロードが可能です。これらのライブラリーは、本製品の使用には必須ではないことに注意してください。

MPEG-1、MPEG-2、MPEG-4、H.263、H.264、MP3、DV SD/25/50/100、VC-1、G.722.1、G.723.1A、G.726、G.728、G.729、GSM/AMR、GSM/FR、JPEG、JPEG 2000、Aurora、TwinVQ、AC3 および AAC は、ISO、IEC、ITU、SMPTE、ETSI およびその他の組織によって制定されている国際標準規格です。これらの標準規格の実装、または標準規格対応のプラットフォームの使用には、インテルを含むさまざまな組織からのライセンス許諾が必要になる場合があります。

Intel、インテル、Intel ロゴ、Intel Core、Itanium、Pentium、Xeon は、アメリカ合衆国およびその他の国における Intel Corporation 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。

© 2010 Intel Corporation. 無断での引用、転載を禁じます。