2007 年 6 月

インテル® コンパイラー バージョン 10.0 最新機能

菅原清文
ビジネス・デベロップメント・マネージャー
ソフトウェア製品部
インテル株式会社

公開が禁止された情報が含まれています。
本資料に含まれるインテル® コンパイラー 10.0 についての情報は、米国時間 6 月 5 日まで公開が禁止されています。

詳 細 情 報

優れた数々の機能を紹介…

コンパイラー 10.0 のスタティックの検証（SV）

スタティックの検証機能は、コンパイル時またはリンク時に、アプリケーション全体に渡り、ユーザーコード内のさまざまな不具合や言語機能と矛盾している箇所を特定

- **C++ および Fortran ユーザーによる開発、そして、メモリーアクセス、バッファー・オーバーフロー、OpenMP* 使用問題についてのデバッグをサポート**
- **OpenMP API およびデータ依存性の問題を発見**

初/中級開発者にとってのメリット:

- 200 種類以上の潜在的なコーディング・エラーを検出
- C/C++ および Fortran 言語の機能を正しくユーザーに「指導」（例: OpenMP 宣言子）

上級開発者にとってのメリット :

- 発見が困難なエラーを検出（例: タイプミス、初期化されていない変数）
- 異なるプログラム単位中の矛盾するオブジェクト宣言の検出（例: 異なるデータ型の仮引数や実引数）

```
$ cat -n main.cpp
     1  #include <iostream>
     2  int foo(const char *);
     3  int main() {
     4     char * y=NULL;
     5     std::cout << foo(y);
     6     return(0);
     7  }
```

```
$ cat -n test.cpp
     1  #include <string>
     2  int foo(const char * widget) {
     3    return(std::strlen(widget));
     4  }
```

```
$ icc -diag-enable sv3 main.cpp test.cpp
main.cpp(5): error #12143: [SV] "y" is
uninitialized
test.cpp(3): warning #12086: [SV] header-file
containing the declaration of intrinsic "strlen"
should be included or forward declaration is
wrong
```

コンパイラー 10.0 の Mudflap サポート

危険なポインターの操作は、コンパイラーによりインストルメントされ、バッファー・オーバーフローや不正なヒープの使用を防ぎます。

```
$ cat -n of-calc.c
 1  #include <math.h>
 2  void calcSqrt(double *a, int N){
 3    for (int i=1;i<N;i++)
 4      a[i]=sqrt( (double)i );
 5    return;
 6  }
$ cat -n of-main.c
 1  #include <stdio.h>
 2  void calcSqrt(double *a, int N);
 3  int main() {
 4    double a[10];
 5    for (int i=0;i<10;i++) {
 6      a[i] = (double) i;
 7    }
 8    calcSqrt(a,11);
 9    return 0;
10  }
```

```
$ icc –o t-mudflap of-calc.c of-main.c -fmudflap
–lmudflap

$ ./t-mudflap
*******
mudflap violation 1 (check/write):
time=1177545175.621017 ptr=0x7fff5bc461d0
size=8
pc=0x2aaaaaae74a1 location=`of-calc.c:4
(calcSqrt)'

/usr/lib64/libmudflap.so.0(__mf_check+0x41)
[0x2aaaaaae74a1]
      ./t-mudflap(calcSqrt+0x8a) [0x40187c]
      ./t-mudflap(main+0x3b5) [0x401c5d]
Nearby object 1: checked region begins 1B after
and ends 8B after
mudflap object 0x136ad320: name=`of-
main.c:4 (main) a‘
bounds=[0x7fff5bc46180,0x7fff5bc461cf]
size=80 area=stack check=0r/10w liveness=10
alloc time=1177545175.621003
pc=0x2aaaaaae6fc1
```

HPO – ハイパフォーマンス・パラレル・オプティマイザー

IA32/インテル® 64/ IA64 システム向けの新しいパラレライザーおよびベクトライザー

- 以下のオプションを使用
  - -O3 –Q[a]x[P,T,N..]
  - -parallel –O3
- まったく新しいデザイン – HLO の後のベクトライザー (「ループ分配」のようなループの最適化)
  - IA32 およびインテル® 64 システムでループ変換を有効にし、ベクトル化。または、IPF でロードペア生成。
- 各最適化間のより良い相互作用
- 最適化されたシリアルコードよりも優れた並列化
- より効率的なマルチスレッド・コード
- 強化されたループ変換
- 1 つになったベクトル化と並列化のコストモデル

HPO はパフォーマンス、特にシリアルおよびパラレル実行のループを向上させる

HPO の例

```
subroutine matmul(a,b,c,n)
real(8), dimension(n,n) :: a,b,c

c=0.d0                                  ! 4
!$omp parallel do                       ! 5
do i=1,n                                ! 6
  do j=1,n                              ! 7
    do k=1,n                            ! 8
      c(j,i)=c(j,i)+a(k,i)*b(j,k)
    enddo
  enddo
enddo
end
```

最適化レポート (簡略)

```
matmul.f90(5): (col. 7) : OpenMP DEFINED LOOP WAS PARALLELIZED
High Level Optimizer Report (matmul_)
matmul.f90(4): (col. 1) : LOOP WAS VECTORIZED.
matmul.f90(7): (col. 3) : PERMUTED LOOP WAS VECTORIZED
LOOP INTERCHANGE in loops at line: 7 8
Loopnest permutation ( 1 2 3 ) --> ( 1 3 2 )

Block, Unroll, Jam Report:
(loop line numbers, unroll factors and type of transformation)

Loop at line 7 blocked by 111
Loop at line 8 blocked by 111
Loop at line 6 blocked by 111

Loop at line 6 unrolled and jammed by 4
Loop at line 8 unrolled and jammed by 4
```

新しい C++ 例外処理

まったく新しいデザイン

簡略化された内部表示によって、より多くの最適化の可能性を提供:

- 向上した EH 表現
- EH により向上したインライン展開
- 向上した EH 制御フローモデル
- 向上した最適化の可能性。定数の伝播のような以前は最適化が無効にされた箇所など
- gcc* C++ に対して向上したコードサイズ/データサイズ
- より多くの並列化の可能性
- 新しいオプション: -f[no-]exceptions
  - 「ダミー」スイッチとして使用

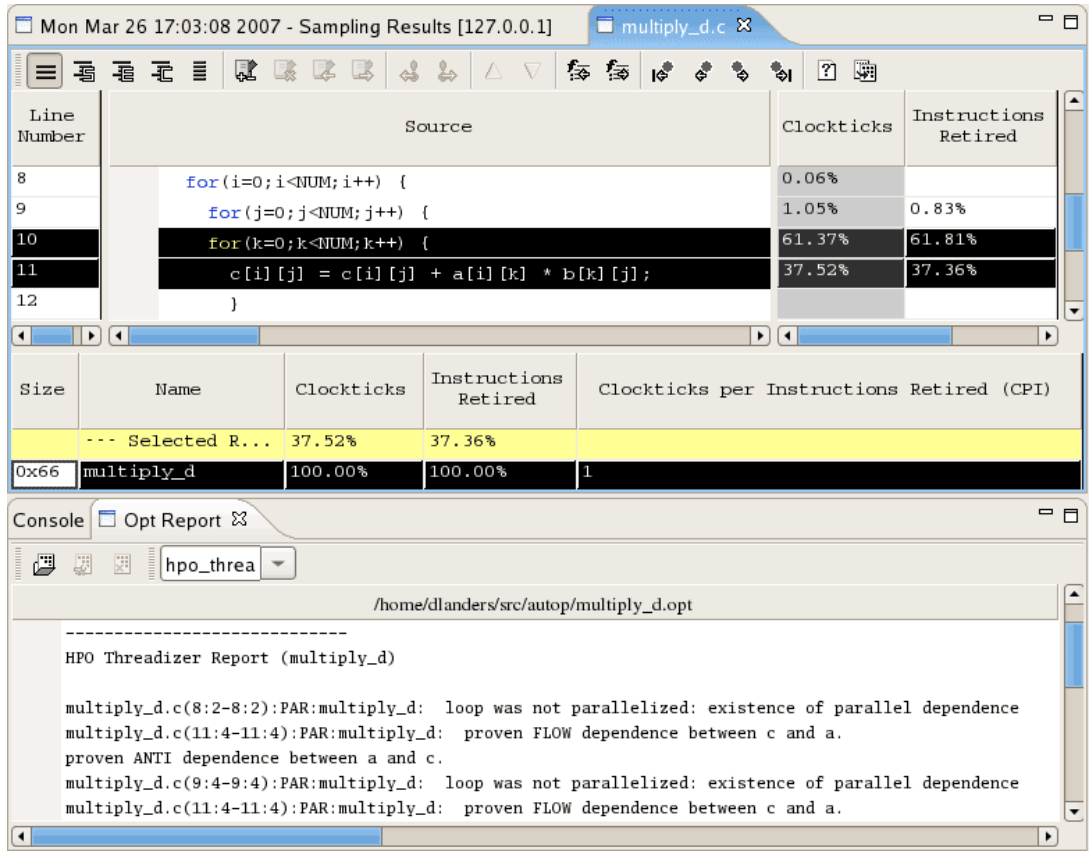
Fortran – 2003 の新機能

- C との互換性保持
  (組み込み関数モジュール ISO_C_BINDING および BIND(C) 属性)
  - 移植性の高い言語混在アプリケーションの作成を可能に
- ISO_FORTRAN_ENV 組み込み関数モジュール
  - 移植性の優れたコーディングを支援
- 非同期 I/O
  - 書式なしデータの読み込み、書き出しランタイム・パフォーマンスの向上
- std03 デフォルトオプション
  - Fortran 2003 規格例外の診断
- プログラマーの作業効率をアップする追加の機能
  (詳細は、リリースノートを参照してください)

インテル® C++ / Fortran コンパイラー 最適化レポート

すばやく正確に診断するのに必要な情報を提供:

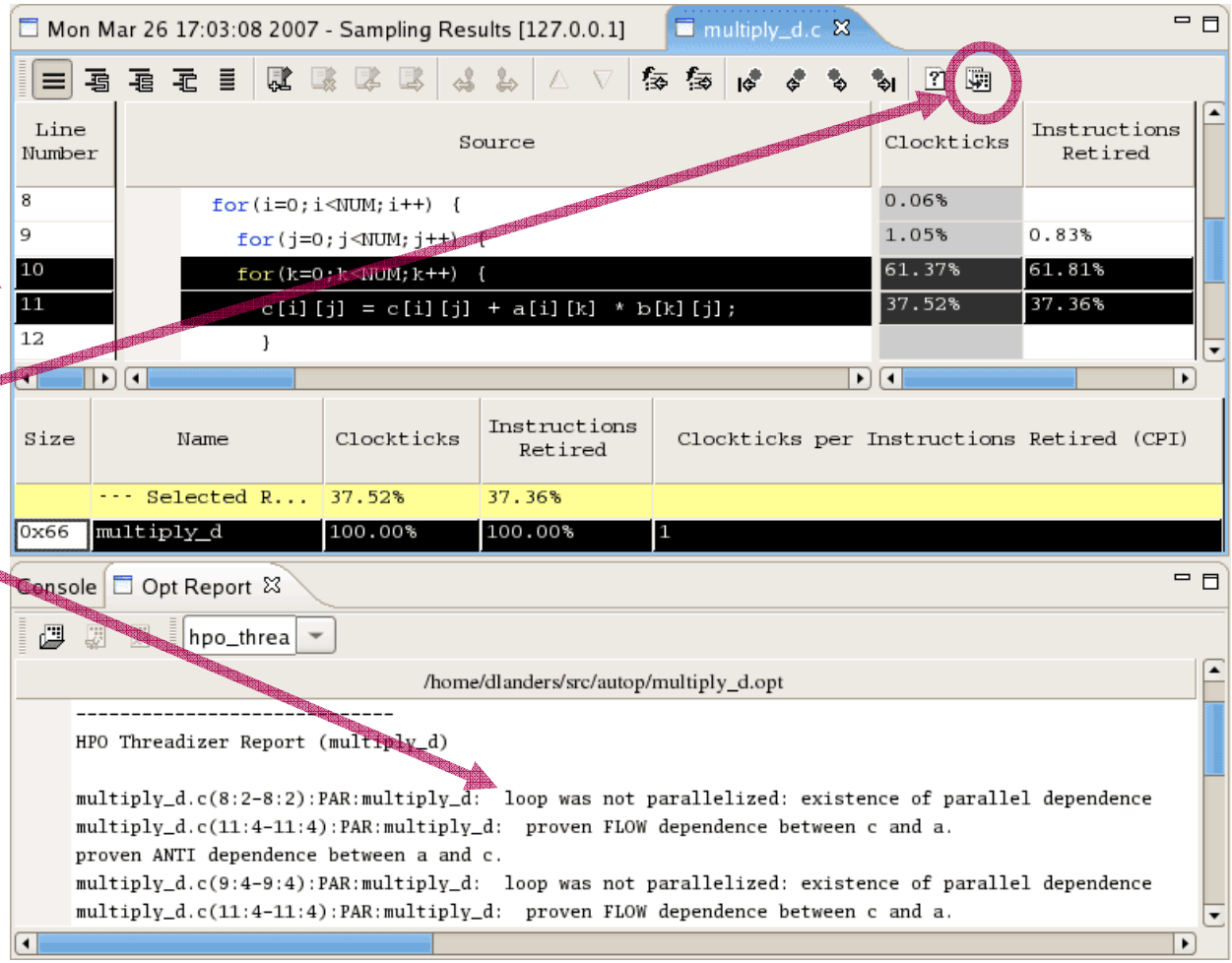
- いつコンパイラーが、選択された高度な最適化を達成したか、またはしなかったかを示す
- アプリケーションのランタイム・パフォーマンスを向上させる箇所を識別
- インテル® VTune™ パフォーマンス・アナライザー Linux* 版により、コード中の hotspot に関するメッセージをハイライト



VTune™ アナライザー Linux* 版

コンパイラー情報を表示

1. インテル® コンパイラーでコンパイル
2. インテル® VTune™ アナライザーを使用して、hotspot を検出
3. ソースビューでコード行を選択
4. 丸で囲まれたアイコンをクリックして、コンパイラーの最適化レポートを表示
5. レポートにより、想定される依存性が原因でコンパイラーが並列化を行わなかったことが判明
6. 依存性がないとわかっている場合は、OpenMP* 文を挿入
7. より高速な並列ソフトウェアが完成
8. インテル® スレッドチェッカーでロジックを確認



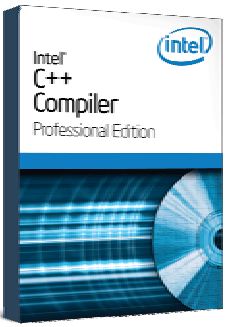
インテル® コンパイラー 10.0

- Windows*版、Linux* 版、Mac OS* 版
- **コンパイラーとライブラリーが一緒に – 各製品はマルチコア並列処理と** ベクトル(SSE)**の並列処理用にチューニング**
- **革命的な最適化機構により、インテルは常にトップに**
  - **コンパイラー *および* ライブラリーの支援**
  - **ライブラリーおよびコンパイラーは常にトップ**
- C++ **並列化: インテル® スレッディング・ビルディング・ブロックが** C++ **製品のスタンダードに**
- Fortran: **優れた新機能**
- **数多くの脆弱性検出機能**

代理店よりお求めいただけます。
評価版を是非お試しください: *www.intel.co.jp/jp/software/products/*

重要: 6 月 5 日までインテル® コンパイラー 10.0 について公開することは禁止されています。

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



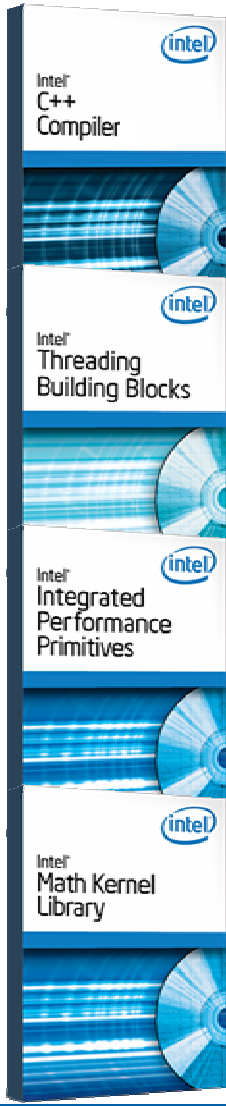
Express your Parallelism with Intel® C++ Compiler Professional Edition

Source Code

Intel® C++ Compiler Professional Edition

Intel® C++ Compiler	Intel® Threading Building Blocks	Intel® Integrated Performance Primitives	Intel® Math Kernel Library
Optimization & Threading OpenMP*, Auto-Parallelization, Vectorization, PGO, IPO & HPO Optimization *Security* Stack frame runtime error checking Static verifier for buffer overflow and OpenMP API verification GNU Mudflap (Linux* / Mac OS*)	*C++ Library for Parallelism* Parallel Algorithms, Containers, Synchronization Primitives Task Scheduler, Memory Allocation	*Optimized Functions for Multimedia* Audio, Video, Imaging, JPEG Speech, Data Compression Signal Processing, Cryptography	*Optimized Functions For Math Processing* BLAS, LAPACK, Sparse Solvers, Fast Fourier Transforms, Vector Math, Statistics

Highly Optimized Application with Improved Thread Performance and Security



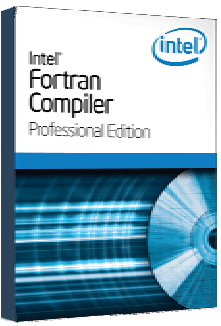
ライブラリーも同梱されるインテル® C++ コンパイラーは、多くの Mac OS* X 開発者に好まれています。

プロフェッショナル・エディションには Windows*版と Linux*版があります。

マルチコア世代の最高の C++ & Fortran 開発ソリューション

重要: 6 月 5 日までインテル® コンパイラー 10.0 について公開することは禁止されています。

インテル® Fortran コンパイラー ・プロフェッショナル・エディション



ライブラリーも同梱されるインテル® Fortran コンパイラーは、多くの Mac OS* X 開発者に好まれています。

プロフェッショナル・エディションには Windows*版と Linux*版があります。

Express your Parallelism with Intel® Fortran Compiler Professional Edition

Source Code

Intel® Fortran Compiler Professional Edition

Intel® Fortran Compiler	Intel® Math Kernel Library
Optimization & Threading OpenMP*, Auto-Parallelization, Vectorization, PGO, IPO & HPO Optimization *Security* Static verifier for buffer overflow and OpenMP API verification	*Optimized Functions For Math Processing* BLAS, LAPACK, Sparse Solvers, Fast Fourier Transforms, Vector Math, Statistics

Highly Optimized Application with Improved Thread Performance and Security

IMSL* ライブラリーは「プロフェッショナル・エディション IMSL 同梱」製品にのみ含まれています。(Windows のみ)

マルチコア世代の最高の C++ & Fortran 開発ソリューション

重要: 6 月 5 日までインテル® コンパイラー 10.0 について公開することは禁止されています。