



インテル® ソフトウェア開発製品による SciMark* 2.0 の最適化

ソフトウェア&ソリューションズ統括部

ソフトウェア製品部

Rev 5/8/2007



Intel, インテル, Intel logo, Itanium は、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。
© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

はじめに

SciMark* 2.0 は米国 NIST (National Institute of Standard and Technology) で開発された浮動小数点演算のベンチマークで、5つの演算カーネルの要素を含む

- FFT (一次元並列高速複素数フーリエ変換)
- SOR (SOR 法による 2 次元ラプラス方程式の解決)
- MC (モンテカルロ法による の演算)
- MV (スパース マトリクス-ベクトル乗算)
- LU (N x N 密行列の LU 分解の演算)

これらの演算は多くの科学技術演算で利用されるため、システムの性能を求める指標として利用される

8-2

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel logo は、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

SciMark* で重要な要素

ベンチマーク	Small	Large
FFT	N = 1024	N = 1048576
SOR	100 x 100	1000 x 1000
MV	N = 1000, NZ = 5000	N = 100000, NZ = 1000000
LU	100 x 100	1000 x 1000

SciMark の MC 以外のカーネルは小さいデータと大きなデータの解決を行う。小さなデータは CPU とキャッシュ階層の評価に有効で、大きなデータはメモリーサブシステムのストレス評価に有効

8-3

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

評価で利用するシステムについて

プロセッサ: インテル® Xeon® プロセッサ (3.6GHz 2MB L2 キャッシュ)
マザーボード: インテル サーバー ボード SE7520AF2
BIOS バージョン: P06
メモリー: 512MB DDR2

OS: RedHat Enterprise Linux* AS 3
Linux カーネル: 2.4.21-20.EL #1 SMP

コンパイラー: インテル® C/C++ コンパイラー Linux バージョン 8.1
GNU C コンパイラー gcc* 3.2.3

ライブラリー: インテル® クラスター・マス・カーネル・ライブラリー バージョン 7.2

8-4

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

初期性能測定 – gcc デフォルト・オプション

ベンチマーク	Small 性能 (MFLOPS)	Large 性能 (MFLOPS)
FFT	149	36
SOR	401	393
MC	47	47
MV	213	211
LU	297	292
総合スコア	221	196

8-5

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

性能測定 – gcc による最適化

ベンチマーク	Small 性能 (MFLOPS)	Large 性能 (MFLOPS)
FFT	510	45
SOR	524	495
MC	206	206
MV	857	453
LU	884	392
総合スコア	596	318

-O3 -march=nocona -ffast-math -mfpmath=sse

8-6

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

性能測定 - インテル® コンパイラー による最適化

ベンチマーク	Gcc Small 性能 (MFLOPS)	Icc Small 性能 (MFLOPS)	性能アップ
FFT	510	512	1.0
SOR	524	759	1.4
MC	206	153	0.7
MV	857	883	1.0
LU	884	1282	1.4
総合スコア	596	718	1.2

8-7

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

性能測定 - インテル® コンパイラー による最適化

ベンチマーク	Gcc Large 性能 (MFLOPS)	Icc Large 性能 (MFLOPS)	性能アップ
FFT	45	45	1.0
SOR	495	720	1.5
MC	206	153	0.7
MV	453	455	1.0
LU	392	402	1.0
総合スコア	318	355	1.1

8-8

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

インテル® コンパイラーの最適化オプションを使用する

-fast オプションは幾つかの最適化オプションを有効にする

- O3: デフォルトの最適化に加えループの最適化を行う
- ipo: ファイル間の最適化を行う
- xP: プロセッサ依存の命令セット(SSE3)を有効にする

-fno-alias (/W) : メモリー参照にエイリアシングが無いと仮定する

SciMark 2.0 のソースコードにはメモリー参照にエイリアシングはない

8-9

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

性能測定 - インテル® コンパイラー による最適化

ベンチマーク	Gcc Small 性能 (MFLOPS)	icc Small 性能 (MFLOPS)	性能アップ
FFT	510	512	1.0
SOR	524	1092	2.1
MC	206	447	2.2
MV	857	832	1.0
LU	884	1827	2.1
総合スコア	596	943	1.6

-fast -fno-alias (Linux*/Mac OS*)
/fast /W (Windows*)

8-10

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

性能測定 - インテル® コンパイラーによる最適化

ベンチマーク	Gcc Large 性能 (MFLOPS)	Icc Large 性能 (MFLOPS)	性能アップ
FFT	45	45	1.0
SOR	495	1015	2.1
MC	206	447	2.2
MV	453	457	1.0
LU	392	389	1.0
総合スコア	318	389	1.2

`-fast -fno-alias (Linux*/Mac OS*)`
`/fast /W (Windows*)`

8-11

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

汎用ライブラリーを利用する

大部分のベンチマーク・プログラムは、各種プラットフォームで実行できるようポータビリティを考慮しており、多くの数値演算ライブラリーが利用可能

リニア代数関数の標準インターフェースが提供される

- BLAS (Basic Linear Algebra Subprograms)
- LAPACK (Linear Algebra Package)

インテル® マス・カーネル・ライブラリー

- 高度に最適化された BLAS、LAPACK そしてフーリエ変換関数を提供
- 乱数発生と超越関数用ベクトル・ライブラリー

SciMark の FFT、MC、MV そして LU カーネルでは、多くの場合ソースコードを変更することなくこれらの関数を利用可能

8-12

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

インテル® マス・カーネル・ライブラリー 9.0

高度に最適化されスレッドセーフな関数群

マルチコア対応

スレッドセーフ

マルチプロセッサ・システムで優れたスケーリングを發揮

実行時に自動的にプロセッサを識別

C および Fortran からの呼び出しをサポート

1つのパッケージですべてのインテル® プロセッサに対応

ロイヤリティー無料の再配布権

- ✓ BLAS
- ✓ LAPACK
- ✓ ScaLAPACK
- ✓ スパースソルバー
- ✓ 高速フーリエ変換
- ✓ ベクトル演算

サポート	Windows*	Linux*	Mac OS*	64 ビット	マルチコア	AMD*
インテル® MKL						



「インテル® マス・カーネル・ライブラリーの DGEMM ライブラリーを導入したところ、当社の標準ベンチマーク結果が 43% ~ 71% も高速化しました。これは実に驚くべき結果です。」

Matt Dunbar 氏
ソフトウェア・デベロッパー
ABAQUS, Inc

8-13

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

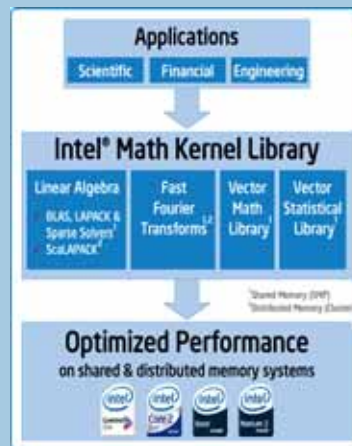
Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

インテル® マス・カーネル・ライブラリー クラスター・エディション 9.0

パフォーマンスを最大限に引き出す高度に最適化された演算ライブラリー

新機能

- 新しいマルチコア・プロセッサ、インテル® Xeon® プロセッサ 5100 番台および 5300 番台向けに最適化
- 新しい VML 関数
 - floor, ceil, round, trunc, hypot など。
- 新しい FMGRES 反復法スパースソルバー
- Fortran と C の FFTW インターフェイス
- 新しい部分微分方程式ソルバー
 - Helmholtz 方程式、Poisson 方程式、Laplace 方程式
- 新しいユーザズ・ガイドと Linux の man ページ



「インテル® マス・カーネル・ライブラリーで特筆すべきは、乱数生成におけるブロック分割オプションです。これは並列アプリケーションで非常に役立ちます。」

Mike Giles 氏
オックスフォード大学教授

8-14

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

インテル® マス・カーネル・ライブラリー を利用したチューニング

8-15

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

インテル® MKL のフーリエ変換 API を利用して FFT カーネルをチューニング

高速フーリエ変換はさまざまなアプリケーションで利用されるアルゴリズムで、開発者は標準化された離散フーリエ変換 (DFT) ライブラリーを利用できる

しかし、DFT 関数では線形代数のように呼び出し規則が標準化されておらず、SciMark 2.0 の FFT カーネルでインテル® MKL を利用するにはコード変更が必要

SciMark 2.0 の FFT カーネルは 1024 個の配列要素に対し、1D フォワード/バックワード変換を行う。インテル® MKL を利用してこの変換処理を行うには、次のようにコードを変更する

8-16

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

インテル® MKL のフーリエ変換 API を利用して FFT カーネルをチューニング (コード変更1)

```
#include <mkl.h>
```

```
Int N = 1024;  
long status;  
double *x = RandomVector ((2 * N), R);  
double scale = 1.0 / (double) N;  
DFTI_DESCRIPTOR *dftiHandle;
```

DftiCreateDescriptor 関数を呼び出し
ハンドルの初期化を行う:
倍精度データ
サイズNのCOMPLEX 1D DFT

```
status = DftiCreateDescriptor (&dftiHandle, DFTI_DOUBLE, DFTI_COMPLEX, 1, N);  
status = DftiSetValue (dftiHandle, DFTI_BACKWARD_SCALE, scale);  
status = DftiCommitDescriptor (dftiHandle);
```

DftiSetValue 関数で後方変換の
スケーリングを設定

```
status = DftiComputeForward (dftiHandle, x );  
status = DftiComputeBackward (dftiHandle, x);  
  
status = DftiFreeDescriptor (&dftiHandle);
```

8-17

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

インテル® MKL のフーリエ変換 API を利用して FFT カーネルをチューニング (コード変更2)

```
#include <mkl.h>
```

```
Int N = 1024;  
long status;  
double *x = RandomVector ((2 * N), R);  
double scale = 1.0 / (double) N;  
DFTI_DESCRIPTOR *dftiHandle;
```

DftiCommitDescriptor 関数を呼び出し
ハンドルを明示化する。以降 dftiHandle
は、ディスクリプターと一致する配列の変換
に利用できる

```
status = DftiCreateDescriptor (&dftiHandle, DFTI_DOUBLE, DFTI_COMPLEX, 1, N);  
status = DftiSetValue (dftiHandle, DFTI_BACKWARD_SCALE, scale);  
status = DftiCommitDescriptor (dftiHandle);
```

```
status = DftiComputeForward (dftiHandle, x );  
status = DftiComputeBackward (dftiHandle, x);  
  
status = DftiFreeDescriptor (&dftiHandle);
```

DftiComputeForward 関数と
DftiComputeBackward 関数
で指定されるディスクリプターによって
変換を行う

ディスクリプターに割り当てられたメモリーを解放

8-18

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

性能測定 - インテル® MKL による最適化

ベンチマーク	Gcc baseline 性能 (MFLOPS)	MKL baseline 性能 (MFLOPS)	性能アップ
Small (N = 1024)	510	1817	3.6
Large (N = 1048576)	45	600	13.3

インテル® MKL はマルチスレッドに対応するスレッドセーフな機能を提供し、マルチコアやマルチプロセッサ搭載のシステムでは優れたスケーリングを発揮

8-19

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

インテル® MKL の LAPACK を利用して LU 分解カーネルをチューニング

LU 分解はフーリエ変換と同様に重要な数値演算処理であり、インテル® MKL の LAPACK でも SciMark 2.0 の LU カーネルで有用な関数を提供する

dgetrf 倍精度 M x N 配列の LU 分解を行う

SciMark 2.0 中のハンドコードされた LU 分解アルゴリズムを dgetrf 関数に置き換えることができるが注意が必要。LAPACK は Fortran インターフェイスのみを提供するが、SciMark 2.0 は C 言語で記述されている

Fortran による関数呼び出しでは引数は参照のみであるが、C では異なる。さらに Fortran では配列要素を行オーダーで配置するが、C では列オーダーで配置する。dgetrf で正しい解答を得るには行並びで引数を提供しなければいけない

dgetrf (&N, &N, A, &N, pivot, &error)

8-20

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

インテル® MKL の LAPACK を利用して LU 分解カーネルをチューニング

Call dgetrf (m, n, A, lda, ipiv, info)

入力引数:

m 整数 (配列Aの列数 $m \geq 0$)
 n 整数 (配列Aの行数 $n \geq 0$)
 A 倍精度 DIM (lda, *)
 lda 整数 (Aの最初の次元)

出力引数:

A 整数 (L と U で上書きされる)
 ipiv 整数 (Pivot インデックス)
 DIM MAX (1, MIN(m,n))
 info 整数 (エラーコード)

ベンチ マーク	Gcc ベースライン	MKL 1 スレッド	MKL 2 スレッド	性能アップ 1 スレッド	性能アップ 1 スレッド
Small	884	1680	N/A	1.9	N/A
Large	392	3837	6646	9.8	16.9

8-21

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

LU 分解カーネルをクラスター向けに並列化する

インテル® MKL ではスレッドレベルの並列化に加え、分散メモリー環境での並列化をサポートするため、MKL のクラスターエディションを提供する

ScaLAPACK Scalable LAPACK, LAPACK の機能を提供

しかし、SciMark 2.0 の LU 分解カーネルでは、Large サイズ (1000 x 1000) の問題解決を行っても分散環境で処理するには小さすぎる。インテル® MKL クラスター エディションの LU 分解カーネルではより巨大なデータを処理することが可能

CMKL を利用する手順は;

1. プロセス GRID の初期化
2. プロセス GRID にまたがって分散処理するため各マトリクスにディスクリプターを作成する
3. dgetrf 関数を pdgetrf 関数に置き換える (p はパラレルを現す)
4. プロセス GRID を開放する

8-22

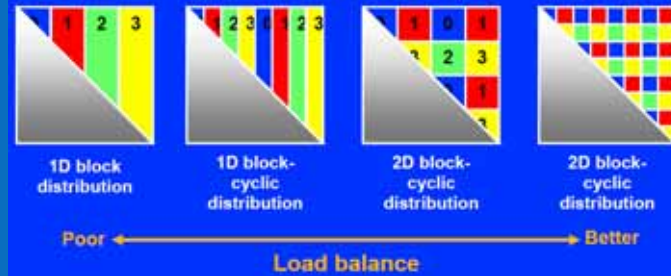
インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

ScaLAPACK によるマトリクス演算の概念



ScaLAPACKにおいて行列は「ブロック周期的」な方式で配布される。行列ディスクリプターで指定するブロックサイズは並列性能に影響する。

大きいブロックサイズはロードバランスに影響を与えるが、通信オーバーヘッドを最小化する。逆に言えば、小さいブロックサイズはロードバランスを改善するけれども、プロセス間の通信を増大させる

8-23

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

ScaLAPACK によるマトリクス演算の実装

```
pdgetrf_ (&global_rows, // グローバルマトリクスの列数
         &global_cols, // グローバルマトリクスの行数
         A, // グローバルマトリクスのサブドメイン
         &one,
         &one,
         descA, // グローバルマトリクスのディスクリプター
         pivot,
         &error );
```

ScaLAPACK ではグローバル配列をローカル配列に分散し、ローカル配列を個々のプロセッサが処理する。他のプロセッサとデータ通信を行い、グローバル配列全体を処理する

ScaLAPACK は MPI 上に構築されているが、ScaLAPACK を利用するにあたり、利用者は MPI ルーチンを直接呼出し制御する必要はない

8-24

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

ScaLAPACK によるマトリクス演算の性能

40000 x 40000 配列の LU 分解をインテル® MKL クラスタエディションで処理

ハードウェア:

インテル® Xeon® プロセッサ 3GHz x 2 搭載する 8 台のシステム
(512KB L2 キャッシュ、ノード毎に 2GB メモリー)
ギガビット・イーサネット + インフィニバンド

性能: 42000 MFLOPS (ギガビット・イーサネット使用)

46000 MFLOPS (インフィニバンド 使用)

8-25

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

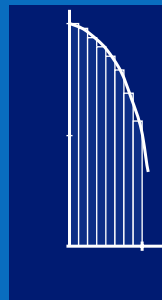
Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

インテル® MKL のベクトル超越関数と OpenMP* を使用した MC カーネルのチューニング

シミュレーションや確立アルゴリズムでは、無作為に決定された乱数の初期状態や継続的なストリーム数を必要とする

インテル® MKL ベクトル統計ライブラリー (VSL) は、最適化された乱数発生器を提供し、均一でない分布の乱数を発生することができる

SciMark 2.0 MC カーネルは、四角形のポイントが無作為にサンプリングしてPIを計算し、円の右上四分円に入っているかどうかを決定します。



8-26

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

SciMark MC カーネルの の計算

```
double MonteCarlo_Integrate (int Num_samples) {
    int under_curve = 0;
    int count;

    Random R = new_Random_seed (SEED);
    for (count = 0; count < Num_smples; count++) {
        double x = Random_nextDouble (R) ;
        double y = Random_nextDouble (R) ;

        if (x*x + y*y <= 1.0) under_curve++;
    }
    Random_delete (R) ;

    Return ((double) under_curve / Num_samples) * 4.0;
}
```

8-27

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

SciMark MC カーネルへの VSL の実装

実装のステップ:

1. VSL から返される乱数ベクトルを格納するため、ブロックサイズ分の配列を定義
2. vsIStream 関数を呼び出し、VSLStreamStatePtr 型の乱数ストリームを初期化する。
3. vdRngUniform 関数の呼び出しにより、“0.0 から 1.0”範囲の均一な分布の倍精度乱数を rnBuf (2 * BLOCK_SIZE) 配列へ格納する
4. 生成した乱数を PI の計算で利用する
5. vsDeleteStream 関数を呼び出し、乱数ストリームを削除する

MC カーネルへの VSL 実装に際し、アルゴリズム自体は変更していない
スカラー版の乱数生成をベクトル版に置き換えている

8-28

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

SciMark MC カーネルへの VSL の実装

```
#include <mkl.h>

double MonteCarlo_Integrate (int Num_samples) {
    int under_curve = 0, i, j, blocks, tail;
    static double rnBuf[2 * BLOCK_SIZE];
    double rnX, rnY; VSLStreamStatePtr stream;

    blocks = Num_samples / BLOCK_SIZE; tail = Num_samples - blocks * BLOCK_SIZE;
    vslNewStream (&stream, VSL_BRNG_MCG31, SEED);

    for (i = 0; i < blocks; i++) {
        vdRngUniform (VSL_METHOD_DUNIFORM_STD, stream, (2 * BLOCK_SIZE), rnBuf, 0.0,
            1.0);
        for (j = 0; j < BLOCK_SIZE; j++) {
            rnX = rnBuf[2 * j]; rnY = rnBuf[2 * j + 1];
            if (rnX * rnX + rnY * rnY <= 1.0) under_curve++;
        }
    }
    vdRngUniform (VSL_METHOD_DUNIFORM_STD, stream, (2 * tail), rnBuf, 0.0, 1.0);
    for (j = 0; j < tail; j++) {
        rnX = rnBuf[2 * j]; rnY = rnBuf[2 * j + 1];
        if (rnX * rnX + rnY * rnY <= 1.0) under_curve++;
    }
    vslDeleteStream (&stream);

    Return ((double) under_curve / Num_samples) * 4.0;
}
```

8-29

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

SciMark MC カーネルの並列化

MC アルゴリズムは並列化できる要素を含んでいる。各乱数サンプルは他のサンプルと独立している。VSL 関数はそれ自体スレッドセーフであるため、MC カーネルを並列化するが、それでは並列化の範囲が狭すぎる

VSL の実装後、OpenMP* を使用して MC カーネルを並列化する

OpenMP:

ポータブルな、共有メモリー型のマルチプロセッシングアプリケーション・プログラム・インターフェイス (API)

- Fortran 77, Fortran 90, C, および C++
- Linux* および Windows* 用の複数のベンダーをサポート

インテル® C/C++ および Fortran コンパイラー バージョン 9.1 では OpenMP の仕様 2.5 をサポート

8-30

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

SciMark 2.0 MC カーネルの VSL / OpenMP* 版

```
nThreads=maxThreads=omp_get_threads();  
omp_set_num_threads(nThreads);
```

```
vslNewStream(&streamX, VSL_BRN_MCG31, SEED);  
vslCopyStream(&streamY, streamX);  
vslLeapfrogStream(streamX, 0, 2);  
vslLeapfrogStream(streamY, 1, 2);
```

乱数ストリームを作成
するための前処理

```
streamXThread=(VSLStreamStausPtr) malloc (nThreads * sizeof  
(VSLStreamStatePtr));  
streamYThread=(VSLStreamStausPtr) malloc (nThreads * sizeof  
(VSLStreamStatePtr));
```

```
for (i=0; i < nThreads; i++) {  
vslCopyStream(&(streamYThread[i]), streamX);  
vslCopyStream(&(streamYThread[i]), streamX);  
vslSkipAheadStream(streamX, BLOCK_SIZE);  
vslSkipAheadStream(streamY, BLOCK_SIZE);  
}
```

各スレッドへ乱数ストリームの
ローカル・コピーを作成

乱数の並びがオーバーラップしないようブロックサイズでスキップ

8-31

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

SciMark 2.0 MC カーネルの VSL / OpenMP* 版

```
#pragma omp parallel for reduction (+:under_curve) ¥  
private(i, j, mX, rnY, threadID, thread_under_curve, rnBufX, rnBufY)
```

8-32

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

SciMark MC カーネル並列化の結果

ベンチマーク	性能 (MFLOPS)	スピードアップ
GNU ベースライン	206	
インテル® コンパイラーのみ	447	2.2
インテル® コンパイラー + VSL	699	3.4
インテル® コンパイラー + VSL + OpenMP*	1003	4.9

8-33

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

インテル® MKL のスパース BLAS を使用した MV カーネルのチューニング

BLAS には行列ベクトル乗算のような共通演算が用意されている
例: dgemv – 倍精度の行列ベクトル積

標準 BLAS に含まれる関数は密行列向けに設定されているが、SciMark 2.0 の MV カーネルは疎行列を用いており大部分の要素はZEROである。疎行列を密行列のように扱い標準 BLAS ルーチンを使用することはできるが、記憶領域と演算パワーを浪費

SciMark 2.0 の MV カーネルは圧縮列格納 (CSR) フォーマットからなる配列を使用: ZERO以外の値を保持する配列と各ZERO以外の値へのカラムインデックスを保持する配列、そして個々の列のZEROではない最初の要素を指すインデックスの配列

8-34

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

インテル® MKL での CSR フォーマット例

9	3	6	3	3	values = (9 3 6 3 3 1 2 5 16)
0	0	0	0	0	columns = (1 2 3 4 5 2 3 4 5)
0	0	1	0	0	rowIndex = (1 6 7 8 9 10)
0	0	0	5	0	
0	0	0	8	16	

•Value と Column インデックス配列のサイズは非ZERO要素の数と等しい

•Row インデックス配列のサイズは、列 + 1 に等しい

•従って、MV カーネルの Large 疎行列は 16MB に格納可能
密行列フォーマットで格納すると 75GB 必要

インテル® MKL では CSR フォーマットをサポートするため、対応する MKL BLAS ルーチンをするに際して、わずかなコード変更で済む

1. mkl_splblas.h ヘッダーをインクルードする
2. インテル® MKL の疎行列インデックスは 1 から始まるため、MV の行と列の配列インデックスも 1 から始まるようにする
3. CSR フォーマット疎配列のベクトル乗算を行うため、mkl_dcsrcgemv 関数を呼ぶ

8-35

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

インテル® MKL のスパース BLAS を使用した MV カーネルのチューニング

ベンチマーク	性能 (MFLOPS)	スピードアップ
GNU ベースライン	183	
インテル® コンパイラーのみ	181	1.0
インテル® コンパイラー + MKL	177	1.0
インテル® コンパイラー + MKL + OpenMP*	362	2.0

N = 1,600,000 および NZ = 250,000,000 のデータサイズは 64 ビットアドレス空間の恩恵を受け、元の MV の Large データと同程度の密度の配列要素を保持するならば、BLAS のスタートアップ・コストを補うことができる

8-36

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

まとめ

- インテル® コンパイラーではソースコードを変更することなくパフォーマンスが向上
- インテル® MKL 用のわずかなコード変更でさらにパフォーマンス向上
- インテル® MKL クラスターエディションは分散コンピューティング環境にも対応
- インテル® コンパイラーがサポートする OpenMP* 機能で容易にマルチコアにも対応
- インテル® MKL はマルチコア環境に対応

8-37

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

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。

© 2007 Intel Corporation.

8-38

インテル® ソフトウェア開発製品



© 2007 Intel Corporation. 無断での引用、転載を禁じます。記載内容は予告なしに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。

Intel, インテル, Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。