

インテル® マス・カーネル・ライブラリー (インテル® MKL) 10.2

詳細

目次	
インテル® マス・カーネル・ライブラリー (インテル® MKL) 10.2	4
特長	4
機能と利点	4
マルチコア対応	4
自動ランタイム・プロセッサ検出機能	4
C および Fortran インターフェイスのサポート	4
すべてのインテル® プロセッサを1つのパッケージでサポート	4
ロイヤルティ無料	4
インテル® MKL 10.2 の新機能	4
パフォーマンス向上	4
C#/.Net サポート	4
BLAS	4
LAPACK	5
FFT	5
PARDISO	5
インテル® MKL 10.1 の新機能	5
計算層	5
PARDISO 直接法スパースソルバー	5
スパース BLAS	5
LAPACK	5
離散フーリエ変換インターフェイス (DFTI)	5
反復法ソルバーのプリコンディショナー	6
ベクトル数学関数	6
ユーザズガイド	6
インテル® MKL 10.2 におけるパフォーマンスの向上	6
インテル® MKL 10.1 におけるパフォーマンスの向上	7
バージョン 10.0 におけるパフォーマンスの向上	7
BLAS	7
LAPACK	7
FFT	7
VML/VSL	7
機能	7
線形代数 - BLAS および LAPACK	7
BLAS	8
スパース BLAS	8
LAPACK	8
BLAS および LAPACK のパフォーマンス	8
線形代数 - ScaLAPACK	9
ScaLAPACK パフォーマンス	9
Raw パフォーマンス	9
ブロックサイズのロバストネス	10
資料	10
線形代数 - スパースソルバー	10
PARDISO*: 並列直接法スパースソルバー	11
新しいアウトオブコア・サポート!	11
反復法ソルバー	11
FGMRES ソルバー	11
共役勾配ソルバー	11
ILU0/ILUT プリコンディショナー	12
スパース BLAS	12

資料	12	LINPACK ベンチマーク	19
高速フーリエ変換 (FFT)	12	使いやすさ	19
インターフェイス	12	パフォーマンス	20
Fortran および C	12	ダウンロード	20
近代的な FFT インターフェイス	12	お問い合わせ/ご意見	20
FFTW インターフェイス	12	資料	20
パフォーマンス (共有メモリー並列処理)	13	互換性	20
2-D 変換	13	オペレーティング・システム	20
3-D 変換	13	開発環境	20
パフォーマンス (分散メモリー)	13	プロセッサ	20
資料	14	テクニカルサポート	20
ベクトル・マス・ライブラリー	15	インテル® プレミアサポート	20
パフォーマンス	15	ユーザーフォーラム	21
精度モード	15	資料	21
資料	16		
ベクトル統計ライブラリー	16		
畳み込みと相関	16		
乱数	16		
基本乱数ジェネレーター (BRNG)	16		
分布ジェネレーター・タイプ	16		
乱数ジェネレーターのパフォーマンス	17		
インテル® MKL と代替ライブラリーの比較	17		
資料	18		

インテル® マス・カーネル・ライブラリー (インテル® MKL) 10.2

インテル® マス・カーネル・ライブラリー (インテル® MKL) は、科学、工学、金融系アプリケーションなど、極めて高いパフォーマンスが求められるアプリケーションに、高度に最適化およびスレッド化された演算ルーチンを提供します。

インテル® MKL はインテル® C++/Fortran コンパイラー・プロフェッショナル・エディション、インテル® クラスタ・ツールキットで利用できるほか、単体製品としても提供されています。

インテル® MKL は、ハイパフォーマンスで将来を保証するアプリケーションを提供し、生産性を向上させます。現在のマルチコア x86 プラットフォーム向けに最適化されていますが、引き続き将来のプラットフォーム向けにも最適化される予定で、最新のアーキテクチャーの強化機能による利点をシームレスにアプリケーションで得ることができます。

Microsoft* Visual Studio* での開発: Microsoft Visual Studio 2003、2005、2008 から BLAS、LAPACK、FFT、統計関数を高速実装するプレミア・ライブラリーを使用して、安定したテクニカル・アプリケーションをより効率良くビルドします。

特長

- バージョン 10.2 リリース! 新機能およびパフォーマンスの向上については、以下のリストをご覧ください。
- インテル® MKL 10.2 は、インテル® Xeon® プロセッサー 5500 番台とインテル® Core™ i7 プロセッサー (開発コード名: Nehalem) 向けに最適化されています。
- サポートされているオペレーティング・システム、コンパイラー、プロセッサーの全リストは、動作環境を参照してください。
- インテル® MKL に関する興味深い新技術については、WhatIf.intel.com (英語) をご覧ください。

機能と利点

インテル® プラットフォーム上で優れたパフォーマンスを発揮

インテル® Xeon® プロセッサー、インテル® Core™ プロセッサー、インテル® Itanium® プロセッサー、インテル® Pentium® 4 プロセッサー・ベースのシステム向けに高度に最適化された演算ライブラリーからリーダーシップ・パフォーマンスを引き出します。特に、新しいクアドコア インテル® Xeon® プロセッサー、クアドコア インテル® Core™ i7 プロセッサー向けのマルチコア・スレッド・パフォーマンスの最適化に重点が置かれています。また、他社製のプロセッサー上でも優れたパフォーマンスを発揮します。

マルチコア対応

- マルチコア、マルチプロセッサー・システムで優れたスケーラビリティを発揮:**
インテル® MKL のビルトインの並列処理を使用して、インテル® Xeon® プロセッサー 5500 番台、および最新のデュアルコア・システム、クアドコア・システムを含むマルチプロセッサー上で優れたパフォー

マンスが得られます。インテル® MKL の BLAS、高速フーリエ変換、ベクトル・マス・ライブラリーおよびその他多くのルーチンが OpenMP* を使用してスレッド化されています。

- スレッドセーフ:** インテル® MKL の関数はすべてスレッドセーフです。スレッド化されていないインテル® MKL の逐次バージョンも提供されています。

自動ランタイム・プロセッサー検出機能

各プロセッサー用に最適化されたコードが実行されるようにランタイムチェックが行われるため、実行しているシステムに関係なく、常に最適なパフォーマンスが提供されます。

C および Fortran インターフェイスのサポート

演算ライブラリーによっては、C/Fortran インターフェイスを使用するために複数の製品を購入しなければならないこともありますが、インテル® MKL では両方のインターフェイスをサポートしています。

すべてのインテル® プロセッサーを 1 つのパッケージでサポート

インテル® MKL は、インテル® Xeon® プロセッサー、インテル® Core™ プロセッサー、インテル® Pentium® 4 プロセッサー、インテル® Itanium® アーキテクチャーを 1 つのパッケージでサポートします。他の演算ライブラリーでは、すべてのプロセッサーをサポートするには複数の製品を購入しなければなりません。

ロイヤルティー無料

インテル® MKL ランタイム・ライブラリーは、開発したソフトウェア製品と一緒に再配布できます (数量無制限)。

インテル® MKL 10.2 の新機能

本リリース (インテル® MKL 10.2) は、最新のインテル® プロセッサー、特にインテル® Xeon® プロセッサー 5500 番台向けに最適化されたマルチスレッド・パフォーマンスを提供します。

パフォーマンス向上

LINPACK、アウトオブコア PARDISO、BLAS、FFT を含むいくつかの主要な演算ルーチンのパフォーマンスが向上しています。さらに、次期インテル® アーキテクチャー・プロセッサーに採用される高度なベクトル化機能向けにインテル® Advanced Vector Extensions (インテル® AVX) もサポートしています。これにより、256 ビットのベクトル演算に対応し、多くの場合でパフォーマンスが 2 倍に向上します。この機能は、アプリケーションにおけるフォワード・スケーリングのテスト、開発を行うために早期に提供されています。

C#/.Net サポート

インテル® MKL 10.2 では、インテル® MKL 関数を呼び出す C#/.Net サポートサンプルが追加されています。

BLAS

レベル 2、レベル 3 BLAS ルーチンのスレッド化のほか、レベル 1 BLAS ルーチンでも幅広くスレッド化が導入されています。

LAPACK

インテル® MKL 10.2 は、LAPACK 3.2 を完全にサポートしています。

FFT

インテル® MKL 10.2 では、以下の機能が FFT へ追加されています。

- 倍率 $1/N$ 、 $1/\sqrt{N}$ のスケーリング

- DFTI_FORWARD_SIGN の実装

- 7、11、13 の混合基数の実装

- クラスタ FFT での実数データ変換の最適化

さらに、FFT 標準をサポートする FFTW インターフェイスが含まれています。

PARDISO

インテル® MKL 10.2 は、PARDISO (Parallel Direct and Iterative Solvers) で単精度をサポートします。また、パフォーマンスも強化されています。

インテル® MKL 10.1 の新機能

インテル® MKL 10.1 は、最新のインテル® プロセッサ (インテル® Xeon® プロセッサ 7400 番台、インテル® Core™ プロセッサ) 向けに最適化されたマルチスレッド・パフォーマンスを提供します。インテル® MKL 10.0 では、新しい「階層化」アーキテクチャーを採用し、ユーザーの多様な使用モデルをより幅広くサポートしています。また、スタンダード・エディションとクラスタ・エディションを結合し、1 つの包括的なパッケージになりました。

新しいインテル® Xeon® プロセッサとインテル® Core™ プロセッサ向けの最適化

詳細は、7 ページの「インテル® MKL 10.1 におけるパフォーマンスの向上」を参照してください。

インテル® MKL 10.0 で採用された「階層化」アーキテクチャー

インテル® MKL 10.0 では、製品を再設計し、複数の階層を提供して、多くのインターフェイス、コンパイラ、プロセッサ構成を 1 つのパッケージでサポートするようにしました。他社のライブラリー製品では、開発環境の特定の構成に応じて製品を見つけ、ダウンロードし、インストールし、テストしなければならないことがあります。この新しいインテル® MKL アーキテクチャーは、インテル® MKL のパフォーマンスを活用するための労力を最小限に抑えながら、多種多様な顧客のニーズに最大限に対応できるよう設計されています。詳細は、インテル® MKL ユーザーズガイドの「インテル® MKL 並列処理の使用」を参照してください。

計算層

このレイヤーはインテル® MKL の中心となる部分です。プロセッサ専用の最適化コードが実行されるようランタイムチェックが実行されます。ユーザーは、必要な専用コードのみが含まれるカスタムの共有オブジェクトをビルドすることができます。そのため、サイズが問題となる場合にこの層のサイズを抑えることができます。

PARDISO 直接法スパースソルバー

- SMP システムで大規模問題を解くアウトオブコア・メモリーの実装
- DSS/PARDISO の前進/後退代入を個別にサポート
- DSS インターフェイスの反復改善をオフにする新しいパラメーター
- PARDISO インターフェイス向け疎行列構造の検証機能のための新しいパラメーター
- スパースソルバー機能が、コアのマス・ライブラリーに統合。別途ソルバー・ライブラリーにリンクする必要がなくなりました。
- スパースソルバー機能に動的にリンクが可能

スパース BLAS

- 圧縮スパース行形式で格納される 2 つの疎行列の和と積を計算するルーチンが追加
- 異なる疎行列形式を変換するルーチンが追加
- すべてのデータ型 (単精度、複素数、倍精度複素数) のサポートの追加
- スパースのゼロベースのインデックスの追加
- 単精度サポートの追加
- レベル 3 スパース BLAS 三角ソルバーのスレッド化

LAPACK

- コールバック関数メカニズムを介した冗長な LAPACK 計算の追跡、中断機能が追加。mkl_progress という関数をユーザー・アプリケーションで定義して、MKL LAPACK ルーチンのサブセットから呼び出すことができます。この機能がサポートされている LAPACK 関数を確認するには、各関数の説明を参照してください。

離散フーリエ変換インターフェイス (DFTI)

- DftiCopyDescriptor 関数が FFT の使用時の便利性的ために追加
- DFTI を呼び出す、静的にリンクされた実行ファイルのサイズが大幅に減少
- 複素数格納が実数-実数変換で利用可能

反復法ソルバーのプリコンディショナー

- インテル® MKL RCI 反復法ソルバーの ILUT アクセラレーター/プリコンディショナー

ベクトル数学関数

- 新しい Mul 関数、Conj 関数、MulbyConj 関数、CIS 関数、Abs 関数
- 数学関数の不正確性がパラメーターの不正確性を決定付けないアプリケーション向け (例: モンテカルロ・シミュレーションやメディア・アプリケーション) に新しい「パフォーマンス強化」モード、EP モードが追加
- すべての VML 関数がスレッド化
- 最適化されたバージョンの正規累積分布関数 (CdfNorm)、逆関数 (CdfNormInv)、逆エラー補完関数 (ErfcInv) がベクトル・マス・ライブラリーに追加

ユーザーズガイド

- インテル® MKL ユーザーズガイドが大幅に向上。インテル® MKL での作業に欠かせないガイドです。ドキュメント・ページからダウンロードできます。
- コンパイラー・サポート: 新しいインテル® コンパイラー 11.0 と PGI* コンパイラーを含む新しいコンパイラーのサポート

インテル® MKL 10.2 における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 プロセッサー上でバージョン 10.1 Update 1 と比べ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® プロセッサー 7400 番台システム上でバージョン 10.1 Update 1 と比べ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 プロセッサー上でバージョン 10.1 Update 1 と比べ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® プロセッサー 7400 番台システム上でバージョン 10.1 Update 1 と比べてパフォーマンスが最大 16-40 倍向上
- インテル® Core™ i7 プロセッサー、インテル® Xeon® プロセッサー (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® プロセッサー 7460 で、大規模な問題の LU 因数分解 (DGETRF) がバージョン 10.1 Update 1 と比べて 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 プロセッサー上で最大 3%、インテル® Core™2 プロセッサー 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 プロセッサー)
 - 3 スレッドで最大 50% の向上 (インテル® Xeon® プロセッサー 7400 番台)
- 非素数サイズのスレッド化 1-D 複素数-複素数 FFT
- 3-D 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下まで減少
- 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 プロセッサーとインテル® Core™2 プロセッサーで次の 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() 関数の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。

インテル® MKL 10.1 におけるパフォーマンスの向上

ライブラリーのあらゆる場所でパフォーマンスが向上しています。次に計測されたいくつかのパフォーマンス・ゲインをリストします。インテル® MKL の各製品ドメイン (BLAS/LAPACK、FFT、VML、VSL など) についてパフォーマンスが向上しています。

- BLAS
- 32 ビットの向上
 - クアッドコア インテル® Xeon® プロセッサ 5300 番台で (Z,C) GEMM が最大 50% 向上
 - クアッドコア インテル® Xeon® プロセッサ 5400 番台で (D,S,Z,C) GEMM コードが 10% 向上
- 64 ビットの向上
 - インテル® Core™ i7 プロセッサで SGEMM が 50% 向上
 - インテル® Core™ i7 プロセッサで DTRSM の右辺のケースが 30% 向上
- 直接法スパースソルバー (DSS/PARDISO):
 - アウトオブコア (OOC) PARDISO が平均で 35% 向上
- VML および VSL
- インテル® Core™ i7 プロセッサでの最適化:
 - 次の VML 関数で最大 17% 向上: Asin, Asinh, Acos, Acosh, Atan, Atan2, Atanh, Cbrt, CIS, Cos, Cosh, Conj, Div, ErfInv, Exp, Hypot, Inv, InvCbrt, InvSqrt, Ln, Log10, MulByConj, Sin, SinCos, Sinh, Sqrt, Tanh
 - 一様乱数生成が最大 67% 向上
 - Wichmann-Hill, Sobol, および Niederreiter BRNG に基づく VSL 分布生成器が最大 10% 向上 (64 ビットのみ)

バージョン 10.0 におけるパフォーマンスの向上

BLAS

- DGEMM のスレッド化が小・中規模のサイズで向上 - 外積サイズ 10%、2 乗サイズ 80%
- DGEMM/SGEMM の大規模 2 乗サイズおよび大規模外積サイズ が 4-5% 向上 (1 スレッド)、10-15% 向上 (8 スレッド)
- DTRSM、DTRMM、DSYRK が 5-30% 向上
- その他のレベル 3 の実関数が大規模サイズで 2-4% 向上

LAPACK

- いくつかの 1 次方程式ソルバー (?spsv/?hpsv/?ppsv, ?pbsv/?gbsv, ?gtsv/?ptsv, ?sysv/?hesv) のパフォーマンスが劇的に向上。帯格納形式および圧縮格納形式で複数の右辺を持つ場合に最大 100 倍に高速化
- 三重対角化ルーチン (?sytrd/?hetrd) が 4 倍に高速化したことにより、すべての対称固有値問題 (?syev/?syev, ?syevd/?heevd, ?syevx/?heevx, ?syevr/?heevr) が大幅に向上
- 圧縮格納形式の三重対角化ルーチン (?sptrd/?hptrd) が 3 倍に高速化したことにより、圧縮格納形式のすべての対称固有値問題 (?spev/?hpev, ?spevd/?hpevd, ?spevx/?hpevx) が大幅に向上
- 直交/ユニタリ変換を適用する多くのルーチン (?ormqr/?unmqr, ?ormr/?unmrq, ?ormql/?unmql, ?ormlq/?unmlq) が最大 2 倍の向上

FFT

- 2 の累乗サイズの複素数 1D FFT において 1 つのスレッドでパフォーマンスが最大 1.8 倍向上
- 64 ビット・モードで実行しているインテル® 64 アーキテクチャー・ベースのシステムで、 2^{22} よりも大きなサイズの単精度複素数の逆方向 1D FFT において 4 スレッドで最大 2 倍、8 スレッドで最大 2.4 倍の高速化 (インテル® Itanium® プロセッサ)

VML/VSL

- 他社製のプロセッサ上で、VSL 関数のパフォーマンスが平均で約 2 倍に向上
- 他社製のプロセッサ上で、VML の vdExp, vdSin, および vdCos 関数が平均で約 18% 向上
- IA-32 およびインテル® 64 アーキテクチャー上で、VSL 関数が平均で約 7% 向上

機能

線形代数: BLAS および LAPACK

BLAS および LAPACK ルーチンは、インテル® プロセッサ用に最適化されており、従来のインプリメンテーションよりもパフォーマンスが大幅に向上しています。インテル® MKL 10.2 は、LAPACK の新リリース 3.2 と互換性があります。

BLAS ライブラリーと LAPACK ライブラリーは、大規模な各種の線形代数問題を解く、長年にわたる標準ライブラリーです。インテル® MKL には、インテル® プロセッサ向けに高度に最適化された BLAS および LAPACK の実装が含まれています。インテル® MKL は、ほかの BLAS と LAPACK の実装に比べて、大幅なパフォーマンス向上を実現します。

BLAS

Basic Linear Algebra Subprograms (BLAS) は、多くの線形代数問題に内在する基本的なベクトル/行列演算機能を提供しています。インテル® MKL の BLAS サポートには下記が含まれています。

- BLAS レベル 1 - ベクトル-ベクトル演算
- BLAS レベル 2 - ベクトル-行列演算
- BLAS レベル 3 - 行列-行列演算
- スパース BLAS - BLAS レベル 1、2、3 の拡張

複数の行列格納スキーム (フル、圧縮、帯) が BLAS レベル 2 と 3 で提供されています。

アプリケーションを変更しなくても、インテル® MKL の並列化 (スレッド化) BLAS ルーチンを使用すると、マルチプロセッシングによるパフォーマンス向上が得られます。アプリケーションでスレッド化を管理する場合でも、インテル® MKL のすべての BLAS 関数はスレッドセーフです。

スパース BLAS

データのスパース性向けに慎重に最適化されたスパース BLAS ルーチンによってパフォーマンス向上と低メモリ要件を実現しています。スパース BLAS には、疎データ (要素の多くがゼロのデータ) 上で一般的なベクトル/行列演算を行う関数セットが含まれています。スパース BLAS は、倍精度と実関数用の選択されたレベル 1、2、3 BLAS ルーチンを対象としています。スパース BLAS は、しばしばスパースソルバーと併せて使用されます。インテル® MKL は、NIST* 式と SparseKit* 式のインターフェイスの両方をサポートしています。

次の行列の型とデータ格納形式がサポートされています。

行列の型	データ格納形式
一般	Compressed Sparse Row (CSR)
対称	Compressed Sparse Column (CSC)
三角	Block Sparse Row (BSR)
対角	Diagonal (DIA)
歪対称	Coordinate (COO) スカイライン

LAPACK

インテル® MKL では、以下の問題を解く Linear Algebra Package (LAPACK) ルーチンが含まれています。

- 線形方程式
- 固有値問題
- 最小二乗問題
- 特異値問題

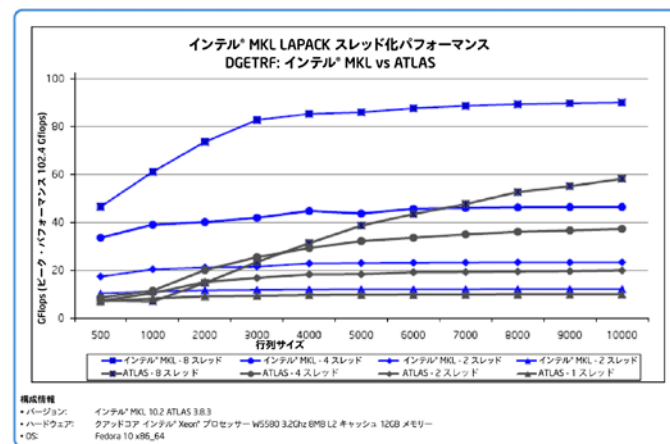
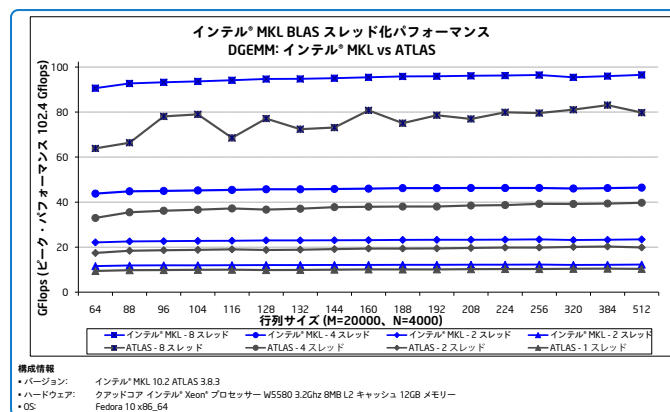
LAPACK ルーチンは、実数データと複素数データの両方をサポートします。ルーチンは、一般、帯、対称またはエルミート、三角、三重対角の行列型の連立方程式がサポートされています。インテル® MKL の LAPACK ルーチンは、複数の行列格納スキームを提供します。LAPACK ルーチンは Fortran インターフェイスで利用可能です。

BLAS および LAPACK のパフォーマンス

倍精度行列積 (DGEMM) は、密線形代数の主ルーチンです。下記のグラフは、インテル® MKL 10.2 と ATLAS* (Automatically Tuned Linear Algebra Software) の DGEMM パフォーマンスを比較したものです。ATLAS は、完全な BLAS API と LAPACK API の小さなサブセットを含む線形代数ソフトウェア・パッケージです。ATLAS の詳細は、<http://math-atlas.sourceforge.net/> (英語) を参照してください。*

インテル® MKL は、パフォーマンスの向上のために最適化され、複数のスレッドで優れたスケーリングを提供するべくチューニングされています。インテル® MKL を今日のデュアルコアやクアッドコア・プロセッサで使用すると、他製品と比べて 2~5 倍のパフォーマンス効果を期待できます。次のグラフは、以下の点を示しています。

1. インテル® MKL は、ATLAS と比べて著しいパフォーマンス効果を達成。
2. インテル® MKL BLAS と LAPACK の目覚ましいマルチプロセッサ/スレッド化のパフォーマンス・スケーリング。



これらのグラフでは、複数のスレッドを使用した場合のインテル® MKL DGEMM および LAPACK (dgetrf) の目覚しいスケーリングと、ATLAS および FFTW のそれぞれに対するパフォーマンスの優位性が見られます。

要約 - インテル® MKL に含まれる BLAS、LAPACK 機能は、インテル® プロセッサ向けに高度に最適化されており、他製品の BLAS、LAPACK 実装と比べてアプリケーションのパフォーマンスを大幅に向上させることができます。

線形代数: ScaLAPACK

ScaLAPACK のインテル® MKL 実装では、標準の NETLIB 実装で大幅なパフォーマンス向上が可能です。

インテル® MKL は、BLAS (PBLAS または Parallel BLAS) の分散メモリ型バージョンとプロセッサ間通信向けの Basic Linear Algebra Communication Subprograms (BLACS) セットを含む ScaLAPACK (Scalable Linear Algebra Package) の基礎コンポーネントを提供します。

ScaLAPACK は、分散メモリ型マルチプロセッサ・マシン (クラスター) 上で線形代数問題を解くためのルーチンを集めた有名な標準パッケージです。

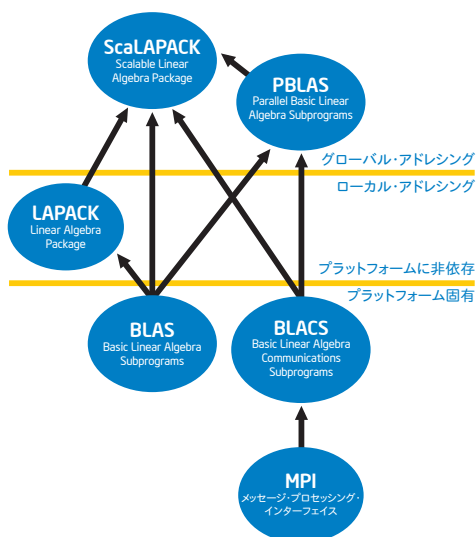


図 1: ScaLAPACK とコンポーネントの関係

ScaLAPACK のダイアグラム。Innovative Computing Laboratory* 提供

ScaLAPACK パフォーマンス

インテル® MKL の ScaLAPACK ライブラリー実装は、特にインテル® Xeon® プロセッサ、インテル® Itanium® プロセッサ、インテル® Pentium® プロセッサ・ベースのシステム向けにチューニングされています。

ScaLAPACK には、直接法ソルバーと固有値問題の 2 つの領域の線形代数が含まれています。そのため、ここでは、PDGETRF (線形連立方程式を解く直接法ソルバー) と PDSYEV (固有値問題を解くのに使用) の両方について考察します。PDGETRF (Parallel, Double precision, GGeneral, TRIangular matrix Factorization) は、行列の多くのクラスに適用できる一般的な因数分解ルーチンであり、また完了する LU 分解

は、線形方程式ソルバーでのパフォーマンスが重要な部分であるため、1 次方程式ソルバーの領域においては重要な関数です。

インテル社で行われたテストでは、インテル® MKL の ScaLAPACK 実装と NETLIB から公開されている実装を比較しました。インテル® MKL と ATLAS の BLAS を使用した Netlib ScaLAPACK のパフォーマンスを示します。ScaLAPACK ライブラリーの詳細は、<http://www.netlib.org/scalapack/> (英語) を参照してください。*

Raw パフォーマンス

図 2 は、64 ビットのインテル® Xeon® プロセッサの 32 ノードのクラスターで各種問題サイズとメモリーサイズにおけるパフォーマンスを示しています。図 2 は、以下の点を例証しています。

1. インテル® MKL の ScaLAPACK は NETLIB ScaLAPACK よりも性能が優れている。
2. インテル® MKL は、ATLAS BLAS を使用した NETLIB ScaLAPACK と比較すると、さらに著しいパフォーマンスを示している。

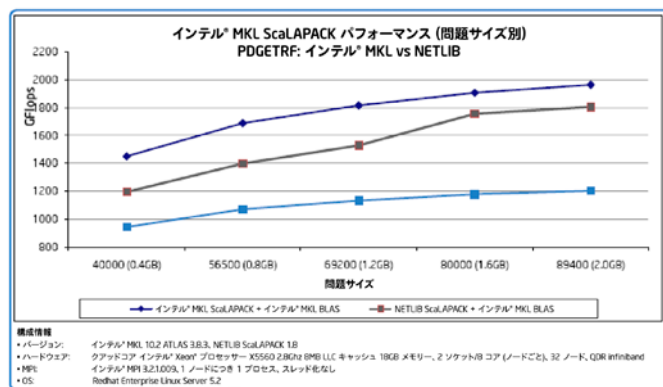


図 2: 各種問題サイズでの PDGETRF パフォーマンス比較

NETLIB ScaLAPACK は、ユーザーが BLAS 実装にリンクする必要があるため、ScaLAPACK と BLAS の最適化によるインテル® MKL のパフォーマンス向上を確認できます。両方ともインテル® MKL BLAS を使用した場合のインテル® MKL と NETLIB の比較では、特に ScaLAPACK の最適化において、インテル® MKL は NETLIB ScaLAPACK 比で 15 パーセントものパフォーマンス効果を達成しています。ATLAS BLAS を使用した場合の NETLIB ScaLAPACK と比較すると、インテル® MKL の ScaLAPACK と BLAS の組み合わせの最適化では、全体でおおよそ 50% のパフォーマンス向上がもたらされています。

図 3 は、実対称行列の固有値と固有ベクトルを計算する PDSYEV についての調査です。同じインテル® Xeon® プロセッサの 32 ノード (64 コア) のクラスターを使用した場合に、NETLIB ScaLAPACK 比でインテル® MKL が 2 倍のパフォーマンスを実現していることがわかります。

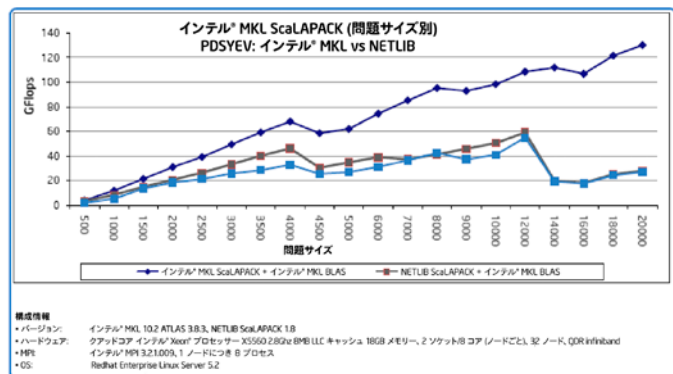


図 3: 各種問題サイズにおける PDSYEV パフォーマンス比較

分散メモリー型並列コンピューティング (クラスター) の大きな利点は、並列コンピューティングにおいて大規模なスケーリングを達成できることです。クラスターユーザーは、システムサイズとともにパフォーマンスもスケーリングするソフトウェア機能に特に関心があります。従来型のテストでは、ノードの増加と比例して問題サイズを大きくし、直線的なパフォーマンス向上をチェックします。図 4 はその結果を示しています。大規模なシステムで ATLAS BLAS を使用した NETLIB に対し、インテル® MKL は著しいパフォーマンス・ゲインを達成していることがわかります。

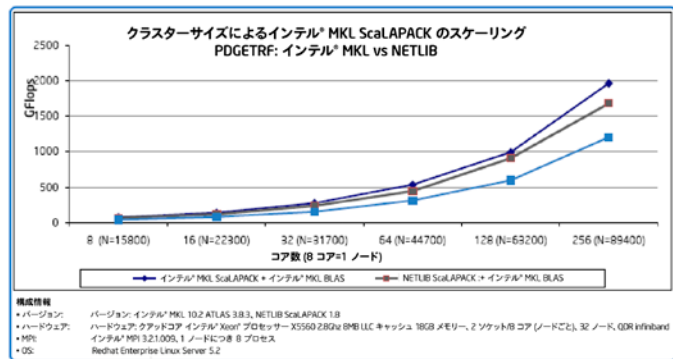


図 4: 各種クラスターサイズにおけるパフォーマンス比較

ブロックサイズのロバストネス

ScaLAPACK を実行する際、データの「ブロック」方法を決定する必要があります。データをノード間に分配する方法を決定するには、適切なブロックサイズの選択が必要です。ブロックサイズは、各ノードに配布されるデータ量を決定します。この作業には労力を要し、誤ったブロックサイズによりパフォーマンスに著しい悪影響をもたらすこともあります。

インテル® MKL の ScaLAPACK 実装はブロックサイズの差異に耐性があります。図 5 は、ブロックサイズに関わらず、インテル® MKL が同様のハイパフォーマンスを達成していることを示しています。NETLIB ScaLAPACK の場合は、ブロックサイズに影響を受けています。

要約 - インテル® MKL の ScaLAPACK 実装は、インテル® プロセッサ向けに高度に最適化されており、その他の ScaLAPACK 実装と比較して、アプリケーションのパフォーマンスを大幅に向上させることができます。

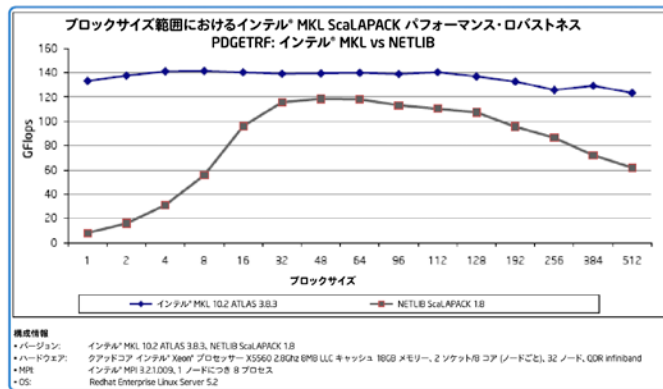


図 5: 各種ブロックサイズにおけるパフォーマンス比較

資料

ScaLAPACK ユーザーズガイド:

ScaLAPACK の使用方法についての詳細な説明は、netlib.org の <http://www.netlib.org/scalapack/slug/node1.html> (英語) を参照してください。

線形代数: スパースソルバー

PARDISO 直接法スパースソルバーは、大規模で疎な線形連立方程式を解くための、使いやすくスレッドセーフでハイパフォーマンスかつメモリー効率の高いソフトウェア・ライブラリーです。バーゼル大学 (University of Basel) からライセンスを許諾されています。インテル® MKL には、共役勾配反復法ソルバーおよび FGMRES 反復法ソルバーも含まれています。

インテル® MKL には、直接的、間接的/反復を使用するスパースソルバーが含まれています。

		インテル® MKL スパースソルバー	
		直接	間接/反復
行列の型			
一般		PARDISO (d, z) (並列直接法ソルバー)	FGMRES (d)
インテル® Core™ プロセッサ・ファミリー	正定値	PARDISO (d, z) (並列直接法ソルバー)	共役勾配 (d)
	不定値	PARDISO (d, z) (並列直接法ソルバー)	

d: 倍精度データのサポート
z: 倍精度データ、複素数データのサポート

PARDISO*: 並列直接法スパースソルバー

新しいアウトオブコア・サポート!

バージョン 10.0 では、PARDISO に対するアウトオブコア・メモリーのサポートが追加されています。コンピューターのメモリー容量は大幅に増加されましたが、依然として、問題サイズが大きすぎるためにインメモリーで解くことができない問題が多数あります。問題サイズの制限に直面している場合は、新しいアウトオブコア PARDISO の解を試してみてください。

バーゼル大学からライセンスを許諾されている PARDISO* ソルバーは、共有メモリー型マルチプロセッサ上で疎な対称、非対称の大規模な連立方程式を解く、スレッドセーフかつハイパフォーマンスなメモリー効率の良いソフトウェア・ライブラリーです。

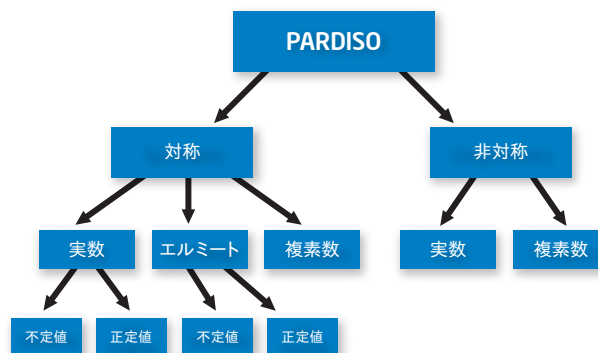
PARDISO ソルバーは、left-looking と right-looking のレベル 3 BLAS スーパーノード手法を組み合わせることで並列化のパイプライン化とメモリー階層を有効に活用します。逐次、並列の疎な数値的因数分解パフォーマンスを向上させるため、アルゴリズムはレベル 3 BLAS Update に基づいています。

十分に大規模な問題サイズの場合、数値実験で並列アルゴリズムのスケラビリティが共有メモリー型マルチプロセス・アーキテクチャーからほぼ独立し、最大 7 倍の速度向上 (8 プロセッサ) が計測されました。この並列化のアプローチは、OpenMP 宣言子に基づいています。

CCLRC (UK) は、直接法スパースソルバーの詳細な分析結果を公開しています。これにより、PARDISO がほかと比べて非常に良いパフォーマンスを発揮することがわかります。レポートは、
<http://epubs.cclrc.ac.uk/work-details?w=34126> (英語)
 を参照するか、または直接 .PDF ファイル [401KB] (<http://epubs.cclrc.ac.uk/bitstream/724/raltr-2005005.pdf>)(英語) を表示してください。

Cranes Software も、有限要素法分析アプリケーションでインテル® MKL PARDISO を使用した論文を公開しています。
<http://www.intel.com/cd/software/products/asmo-na/eng/371766.htm>
 (英語) から .PDF ファイル [187KB] をダウンロードできます。

PARDISO は広範囲の疎行列をサポートしており、共有メモリー型マルチプロセッシング・アーキテクチャーで実数/複素数、構造対称/非対称、正定値/不定値、エルミートのスパース連立線形方程式を解くことができます。



PARDISO スパースソルバー行列

反復法ソルバー

インテル® MKL には、一般および対称正定値線形代数方程式を解く反復法スパースソルバーが含まれています。

ソルバーは、ユーザーが特定の演算 (例えば、行列-ベクトル乗算) をソルバーに提供する必要があるリバース・コミュニケーション・インターフェイス (RCI) スキームをベースとしています。このスキームは、行列-ベクトル乗算などの特定の演算実装に依存しないため、ソルバーに柔軟性を持たせることができます。

FGMRES ソルバー

FGMRES は、疎な一般連立線形方程式を解くよく知られたソルバーです。一般的適用性が高く、幅広い状況に適用できます。ソルバーは行列のスパース性に対応するため、密行列手法で処理可能なサイズより大きな問題サイズを求解でき、また、大規模なスパース問題を密行列法より高速に解くことができます。

共役勾配ソルバー

共役勾配 (CG) ソルバーは、対称正定値行列で表される連立線形方程式の数値解法に適しています。共役勾配法は反復法のため、直接法では処理できない大規模な疎行列に適用できます。大規模な疎行列は、偏微分方程式を数値的に解く際、定期的に発生します。CG ソルバーは、右辺が 1 つの連立方程式用と、右辺が複数の連立方程式用の 2 つのバージョンで実装されています。

ILU0/ILUT プリコンディショナー

プリコンディショナー (「アクセラレーター」とも呼ばれます) は、反復的な解のプロセスを加速するために使用されます。プリコンディショナーを使用することで反復回数を大幅に減らし、より優れたソルバー・パフォーマンスを導く場合もあります。インテル® MKL には、現在 ILU0 (不完全LU分解) と ILUT (しきい値不完全LU分解) の2つのプリコンディショナーが含まれています。ILU0/ILUT プリコンディショナーは任意の非退化行列に適用でき、単体で使用するか、あるいは MKL FGMRES ソルバーと併用します。両プリコンディショナーとも、元の行列を2つの三角行列 (下三角行列と上三角行列) の積に因数分解する、よく知られた因数分解に基づいています。通常、この分解により、元の行列と比べて結果の行列構造にフィルインが導かれますが、ILU0 プリコンディショナーは、構造を元の行列のまま維持します。一方、ILUT プリコンディショナーはフィルイン数を制御します。また、ILUT プリコンディショナーはより高い信頼性を備え、ILU0 プリコンディショナーによる計算は一般にメモリー消費量が少なく、高速です。

スパース BLAS

スパースソルバーは、しばしばスパース BLAS と併せて使用されます。スパース BLAS は、疎データ上で多くの一般的なベクトル/行列演算を行う関数セットです。スパースベクトルと行列は、その要素の大半がゼロです。インテル® MKL には、データのスパース性を有効に利用できるよう特に最適化されたスパース BLAS 実装が含まれています。スパース BLAS は、選択されたすべてのデータ型の BLAS レベル 1 ルーチンと倍精度実関数用のレベル 2、3、3 BLAS ルーチンを対象としています。行列の型には、一般行列、対称行列、三角行列、非対称行列、対角行列などが含まれます。Compressed Sparse Row (CSR)、Compressed Sparse Column (CSC)、対角、座標、スカイライン形式などのデータ構造をサポートしています。

資料

『Linear Solver Basics』(PDF 377KB): 連立1次方程式の解法に関連した用語と概念の概要については、
http://cache-www.intel.com/cd/00/00/22/97/229716_229716.pdf (英語) を参照してください。

『CCLRC Report on Direct Sparse Solvers』(PDF 401KB): 多くの直接法スパースソルバーの詳細なパフォーマンス比較については、
<http://epubs.cclrc.ac.uk/bitstream/724/raltr-2005005.pdf> (英語) を参照してください。

『Utilization of Parallel Solver Libraries to Solve Structural and Fluid Problems』(PDF 187KB): Finite Element Analysis (FEA、有限要素法分析) アプリケーションでのインテル® MKL の PARDISO スパースソルバーの性能を分析した Cranes Software によるホワイトペーパーは、
<http://www.intel.com/cd/software/products/asm-na/eng/371766.htm> (英語) を参照してください。

高速フーリエ変換 (FFT)

インテル® MKL には、簡単に使用できる C と Fortran インターフェイスの多次元 FFT ルーチン (1D から最大 7D) が含まれています。同じ API で分散メモリークラスターをサポートし、最小の労力で大量のプロセッサに負荷を分散することによってパフォーマンスを向上します。インテル® MKL は、FFTW 2.x と 3.0 インターフェイスとの互換性を提供し、現 FFTW ユーザーは簡単にインテル® MKL を既存のアプリケーションにプラグインできます。

フーリエ変換は、デジタル信号処理、画像処理で使用され、また偏微分方程式 (PDE) ソルバーでも使用されます。インテル® MKL の Fast Fourier Transform (FFT) 機能は、インテル® アーキテクチャー・ベースのシステム向けに高度に最適化されています。

インテル® MKL は、分散メモリー型マルチプロセッサ・アーキテクチャー・マシン (クラスター) のサポートも提供しています。

インテル® MKL の高速フーリエ変換機能:

- ハイパフォーマンス
- 優れたマルチプロセッサ・スケーリング
- 多次元サポート (1-D から最大 7-D)
- 混合基数に対応
- 近代的で使いやすいインターフェイス
- FFTW 2.x および 3.x インターフェイス

インターフェイス

Fortran および C

すべての FFT 関数に対して Fortran と C の両インターフェイスが用意されています。

近代的な FFT インターフェイス

インテル® MKL は、使用と保守がより簡単になるよう設計された新しい FFT インターフェイスを提供します。

FFTW インターフェイス

インテル® MKL には、FFTW 2.x と 3.x 互換 API が含まれ、FFTW ユーザーは最小の労力でインテル® MKL を統合できます。インテルの目的は、FFTW ユーザーがインテル® MKL のハイパフォーマンスで質の高い高速フーリエ変換を簡単に使用できるようにすることです。FFTW とは MIT から提供されている FFT パッケージです。テクニカル・ユーザー・ノート

(http://www.intel.com/software/products/mkl/docs/fftw_mkl_user_notes_2.htm (英語)) を参照してください。

- FFTW 3.x インターフェイス

FFTW 2.x インターフェイス

パフォーマンス (共有メモリー並列処理)

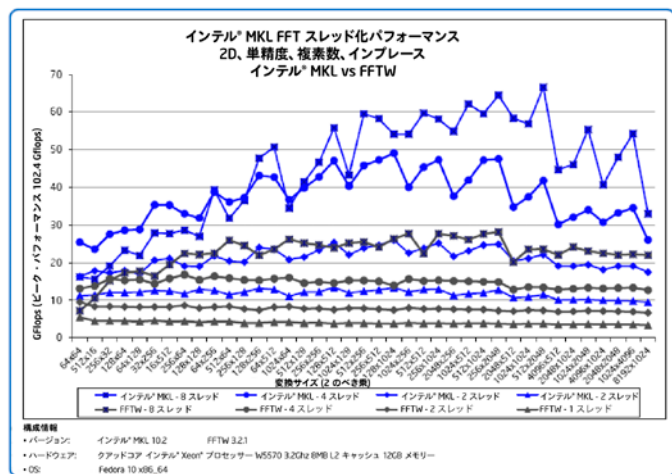
インテル® MKL の高速フーリエ変換は、中・大規模の変換向けに高度に最適化されています。以下のグラフは、インテル® MKL と一般的な FFT ソフトウェア・パッケージの FFTW と比較したものです。このグラフからわかる主な 3 つのポイントは次のとおりです。

- インテル® MKL は中・大規模変換で極めて優れたパフォーマンスを提供している。
- インテル® MKL はマルチプロセッサ・システムで極めて優れたスケラビリティを提供している。
- インテル® MKL は中・大規模変換において FFTW よりも優れている。

注: 以下のすべてのテストでは、FFTW で優れたパフォーマンスを達成できるように FFTW_PATIENT フラグを使用して、fftw_plan_dft_xd() が呼び出されています。

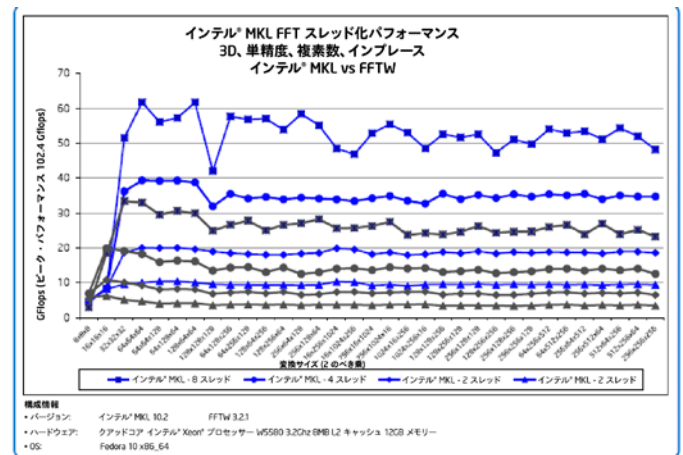
2-D 変換

2 次元変換は、マルチプロセッシングの利点を活用するのに十分な大きさです。次のグラフは、インテル® MKL が FFTW よりも優れたスレッド化パフォーマンスを提供していることを示しています。インテル® MKL のシングル・スレッド・パフォーマンスは、FFTW の 2、4、8 スレッドよりも優れたパフォーマンスを実現していることがわかります。*



3-D 変換

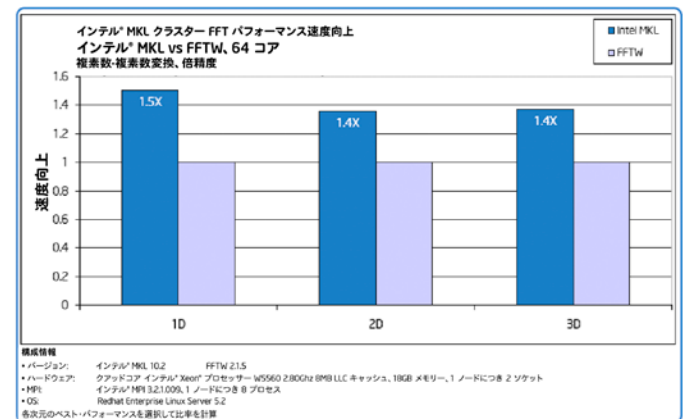
3 次元変換は、2 次元変換と同じ動きを示しています。インテル® MKL は中・大規模変換で極めて優れたパフォーマンスを提供します。*



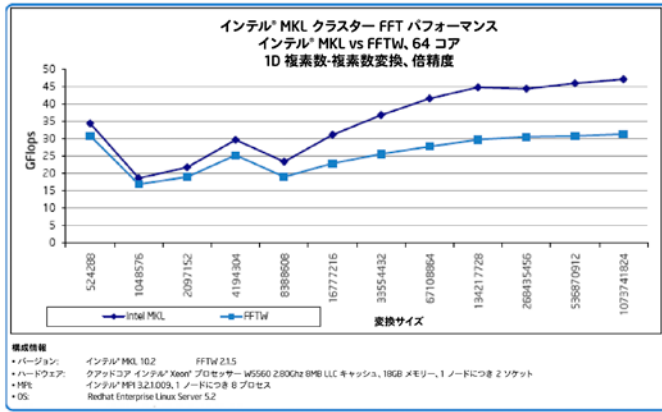
パフォーマンス (分散メモリー)

インテル® MKL は、分散メモリー型マルチプロセッサ・アーキテクチャ・マシン (クラスター) のサポートも提供しています。このサポートには分散型メモリー並列処理向けに設計された FFT が含まれています。クラスターシステムは、共有メモリー型並列システムよりも並列化の度合いが高く構成されているため、インテル® MKL は特に大規模な問題で問題サイズとパフォーマンスのヘッドルームを拡張します。

- インテル® MKL は広範囲の変換サイズで優れたパフォーマンスを発揮している。
- インテル® MKL では問題サイズが大きくなるにつれて、FFTW を上回るパフォーマンス向上が得られている。



インテル® マス・カーネル・ライブラリー (インテル® MKL) 10.2: 詳細



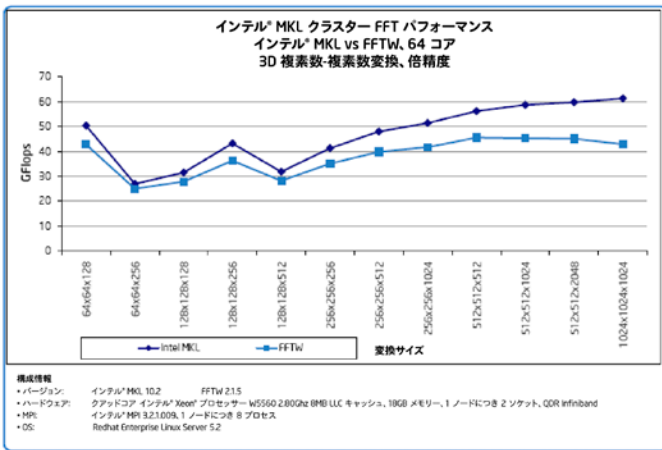
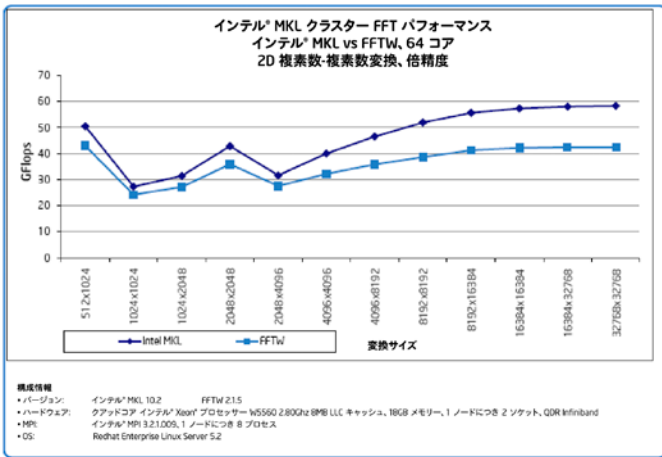
要約 - インテル® MKL の高速フーリエ変換機能は、ハイパフォーマンス・コンピューティング・コミュニティで重要視されている幅広い問題サイズにわたり、極めて優れたパフォーマンスとスケーラビリティを提供します。FFTW とは異なり、インテル® MKL ではソフトウェアを実行する前に、時間のかかるマシン・キャリブレーション実行 (「plan creation」) 処理を行う必要がありません。インテル® MKL は、分散メモリー並列コンピューティング・システムの FFT をサポートし、大規模な問題サイズに対して優れたパフォーマンス・ヘッドルームを提供します。インテル® MKL には FFTW インターフェイスが含まれているため、開発者は簡単にパフォーマンスを比較して、最低限のソースコード変更で FFTW からインテル® MKL に切り替えることができます。

資料

FFTW 2.x ラッパー: FFTW 2.x アプリケーションのインテル® MKL への移植についての詳細は、
http://www.intel.com/software/products/mkl/docs/fftw2xmkl_notes.htm (英語) を参照してください。

FFTW 3.x ラッパー: FFTW 3.x アプリケーションのインテル® MKL への移植についての詳細は、
http://www.intel.com/software/products/mkl/docs/fftw3xmkl_notes.htm (英語) を参照してください。

インテル® MKL と FFTW の API 比較: FFTW とインテル® MKL の高速フーリエ変換 API のマッピングについての記事は、
<http://www.intel.com/cd/ids/developer/asm-na/eng/223902.htm?page=1> (英語) を参照してください。



ベクトル・マス・ライブラリー

ハイレベルな数学関数群 (累乗関数、三角関数、指数関数、双曲線関数、対数関数など) をベクトル化して実装することで、アプリケーションのパフォーマンスを向上します。

インテル® MKL に含まれているベクトル・マス・ライブラリー (VML) は、計算量の多いコア数学関数の高度に最適化されたベクトル実装を提供します。ライブラリーにはすべての VML 関数に対して Fortran と C の両インターフェイスが用意されています。また、すべての関数がスレッド化されています。

サポートされている VML 関数

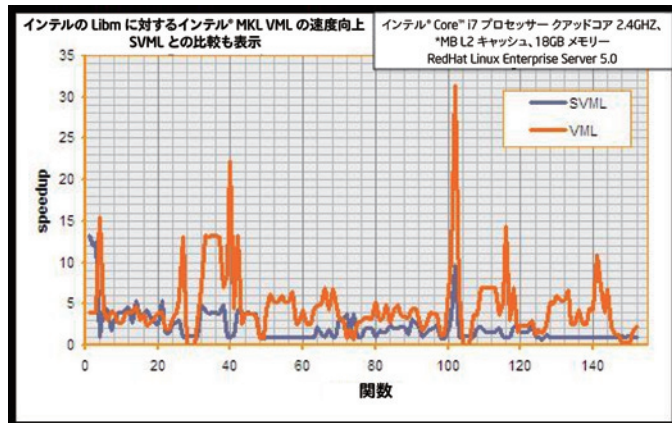
算術	三角	双曲線	累乗/根
Add	Sin [^]	Sinh [^]	Pow [^]
Sub	Cos [^]	Cosh [^]	Powx [^]
Div	SinCos	Tanh [^]	Pow2o3
Sqr	CIS ^{^^}	Asinh [^]	Pow3o2
Mul	Tan [^]	Acosh [^]	Sqrt [^]
Conj ^{^^}	Asin [^]	Atanh [^]	Cbrt
MulByConj ^{^^}	Acos [^]		InvSqrt
Abs	Atan [^]		InvCbrt
	Atan2		Hypot
			Inv

丸め	指数/ 対数	特殊	その他
Floor	Exp [^]	Erf	Inv
Ceil	Expml	Erfc	Div
Round	Ln [^]	ErfInv	
Trunc	Log10 [^]	ErfInV (新規)	
Rint	Loglp	CdfNorm (新規)	
NearbyInt		CdfNormInv (新規)	
Modf			

すべての関数を実数データ型に利用できます。
 ^ は複素数データ型がサポートされていることを示します。
 ^^ は複素数データ型のみサポートされていることを示します。

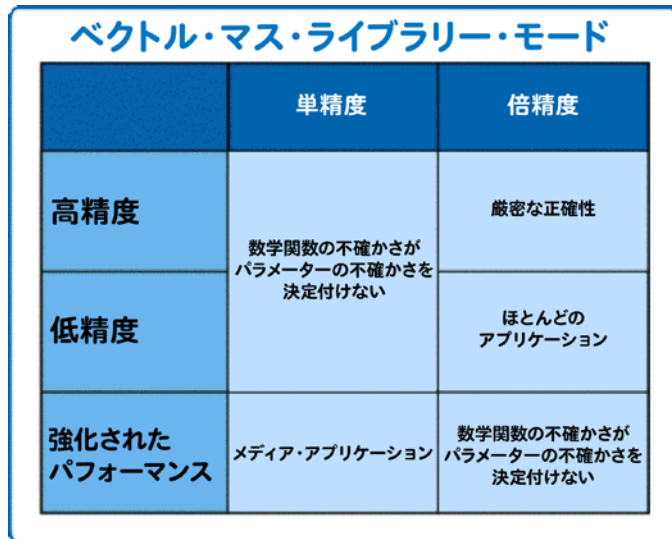
パフォーマンス

インテル® MKL のベクトル・マス・ライブラリーは、スカラー実装よりも多大なパフォーマンスの優位性を提供します。以下のグラフは、インテル® MKL の VML 関数と、インテル® C++/Fortran コンパイラーからの数学関数の呼び出しをサポートするランタイム・ライブラリー LibM および SVML で実装される等価関数を比較したものです。グラフでは、ほとんどの VML 関数が多くの場合、2 倍から 5 倍、いくつかのケースでは最大 15 倍から 30 倍ものパフォーマンス効果が得られていることがわかります。



精度モード

VML 関数は、単精度と倍精度をサポートし、3 つのパフォーマンス/精度モードが提供されています。この複数の精度モードは、インテル® MKL がほかのベクトル化コンパイラーとは一線を画する製品となる重要な機能です。インテル® MKL では、要件に合致する精度を選択することができ、特定のコードのパフォーマンスを最大限に引き出すことができます。



各種インテル® プロセッサベースのシステムにおける全関数の詳細な VML パフォーマンスおよび精度については、
<http://www.intel.com/software/products/mkl/data/vml/vmldata.htm> (英語) を参照してください。

要約 - インテル® MKL のベクトル・マス・ライブラリーを使用すると、コンパイラーによるコードのベクトル化に頼らずにパフォーマンスを最大限に引き出すことができます。インテル® MKL のベクトル・マス・ライブラリーは制御性に優れ、コンパイラーのランタイム・ライブラリーの非ベクトル化スカラー関数と比較して、アプリケーションのパフォーマンスを大幅に向上させることができます。

資料

『Intel® VML Performance and Accuracy』 - 各種インテル® プロセッサにおけるインテル® MKL ベクトル・マス・ライブラリー (VML) のパフォーマンスおよび精度の詳細情報は、
<http://www.intel.com/software/products/mkl/data/vml/vmldata.htm> (英語) を参照してください。

ベクトル統計ライブラリー

インテル® MKL には、サブライブラリーを形成する多くの統計機能が含まれており、これはベクトル統計ライブラリー (VSL) と呼ばれます。このベクトル統計ライブラリーには、畳み込み/相関と広範囲な乱数ジェネレーター群が含まれています。

畳み込み/相関

VSL では、単精度および倍精度データの線形畳み込み/相関変換を実行するためのルーチン群が用意されています。

- フーリエおよび直接アルゴリズム
- 1次元あるいは多次元
- 単精度と倍精度
- C および Fortran インターフェイス
- 近代的で使いやすいタスク志向のインターフェイス
- ESSL ユーザー向け IBM ESSL* インターフェイス

インテル® MKL のベクトル統計ライブラリーの畳み込み/相関ルーチンについての詳細情報は、インテル® MKL リファレンス・マニュアル (<http://www.intel.com/software/products/mkl/docs/WebHelp/mkl.htm>) (英語) を参照してください。

乱数

スカラー乱数ジェネレーターよりも優れたパフォーマンスが得られるベクトル乱数ジェネレーターを使用することで、シミュレーションを高速化します。

インテル® MKL ベクトル統計ライブラリー (VSL) には、さまざまな確率分布用の乱数ジェネレーター群が含まれています。すべての VSL 関数は高度に最適化され、インテル® アーキテクチャー上で極めて優れたパフォーマンスを発揮します。

VSL によりパフォーマンスの著しい向上が期待できるアプリケーションとしては、物理、化学、医療シミュレーションや財務分析ソフトウェアでよく使用されるシミュレーション・アルゴリズムなどがあります。すべての VSL 関数で Fortran と C の両インターフェイスが用意されています。

VSL では、以下に示すように速度と統計的品質が異なる 9 個の基本乱数ジェネレーターが提供されています。

基本乱数ジェネレーター (BRNG)

疑似乱数

MCG59	乗算合同法ジェネレーター 59 ビット
MCG31m1	乗算合同法ジェネレーター 31 ビット
MRG32k3a	複合再帰ジェネレーター 32 ビット
R250	GFSR
Wichman-Hill	273 個の基本ジェネレーター・セット
MT19937	Mersenne Twister
MT2203	1024 個の Mersenne Twister 基本ジェネレーター・セット

準乱数

Sobol	32 ビット Gray コードベース・ジェネレーター
Niederreiter	32 ビット Gray コードベース・ジェネレーター

VSL は、Leapfrog 法およびブロック分割法を含むランダムストリームを作成する複数の手法をサポートしています。大規模なモンテカルロ・シミュレーションの場合、VSL はランダムストリームをファイルに保存したり、ファイルから復元するルーチンを提供します。抽象ストリームは、バッファーに格納された乱数データと分布ジェネレーターの使用において柔軟性を提供します。

VSL はまた、ユーザー定義の基本ジェネレーターおよび多くの連続/離散ジェネレーターをサポートしています。

分布ジェネレーター・タイプ

連続	離散
Uniform	Uniform
Gaussian	UniformBits
GaussianMV	Bernoulli
Exponential	Geometric
Laplace	Binomial
Weibull	Hypergeometric
Cauchy	Poisson
Rayleigh	Poisson (異なる平均)
Lognormal	Negative binomial
Gumbel	—
Gamma	—
Beta	—

乱数ジェネレーターのパフォーマンス

VSL 関数は、単精度、倍精度の実数ベクトル引数で動作し、スカラー実装よりも多大なパフォーマンスの優位性を提供します。下記の表は、インテル® プロセッサ上での VSL 乱数ジェネレーターと乱数分布についての詳細なパフォーマンス情報を示します。

インテル® MKL の最適化パフォーマンスがわかるように、標準 C rand() 関数とインテル® MKL のベクトル様乱数ジェネレーターを比較しています。インテル® MKL の VSL 関数はスレッドセーフのため、OpenMP を使用して並列化された後のインテル® MKL のベクトル様乱数ジェネレーターの速度も提示しました。

VSL パフォーマンスと C コード

インテル® Xeon® プロセッサ・ベースのシステム²

インテル® Xeon® プロセッサ ¹	実行時間 (秒)	高速化 rand() 比 (倍)
標準 C rand() 関数	40.52	1.00
インテル® MKL VSL 乱数ジェネレーター	6.88	5.89
OpenMP* バージョン (8 スレッド)	0.92	44.04

- Two-way のクアッドコア インテル® Xeon® プロセッサ・ベース・システム (合計 8 コア)、2.4GHz、2x8MB L2 キャッシュ、4GB メモリー、Windows® 2003 Enterprise x64 Edition、インテル® C++ コンパイラー 10.0
- インテル® MKL の VSL のインテル® 64 アーキテクチャー・バージョンを計測で使用

インテル® MKL とその他のライブラリーの比較

Visual Numerics Inc. と Numerical Algorithms Group Ltd. はよく知られた数値計算、統計ライブラリーのプロバイダーです。Visual Numerics は、IMSL Fortran 90 MP ライブラリー (F90MP*)、IMSL C Numerical ライブラリー* として知られる Fortran ユーザー向け、C/C++ ユーザー向け数値計算および統計解析サブルーチンを提供しています。Numerical Algorithms Group は、AG Fortran 77 ライブラリー*、NAG Fortran 90 ライブラリー*、NAG C ライブラリー*、NAG SMP ライブラリー*、NAG Parallel ライブラリー* として知られる C/C++ ユーザー向け、Fortran ユーザー向けの数値ライブラリーを提供しています。

次の表は、インテル® MKL 10.2、IMSL F90MP 6.0、NAG Fortran 77 (Mark 21) ライブラリーに備わっている乱数生成機能の概要です。

インテル® MKL 10.2 機能と IMSL*/NAG*

機能	インテル® MKL	IMSL	NAG
BRNGs	7 擬似乱数 2 準乱数	9 擬似乱数 1 準乱数	2 擬似乱数 3 準乱数
ユーザー定義 BRNG	サポートあり	ユーザー定義の BRNG がライブラリー BRNG を置換。ライブラリー・サービスの多くはユーザー定義 BRNG とは動作しない。	サポートなし
シーケンス操作	<ul style="list-style-type: none"> ストリームの作成、コピー、削除のメカニズム ファイルへ/からストリームを保存/格納 1 つまたは複数の BRNG に基づくランダムストリームの任意の数 	<ul style="list-style-type: none"> BRNG を切り替えるサービス・サブルーチン BRNG シードを保存/格納するサブルーチン GFSR とシャッフルされた BRNG のステートテーブルを保存/格納するサブルーチン 	<ul style="list-style-type: none"> BRNG シードを保存/格納するサブルーチン BRNG を切り替えるサービス・サブルーチン
サブシーケンス分割	<ul style="list-style-type: none"> “Skip-ahead” 法と “Leapfrog” 法のサポート Skip-ahead 法はシーケンスの要素の任意の数のスキップが可能 Leapfrog 法は任意の数のオーバーラップしないサブシーケンスに分割が可能 	Skip-ahead 法の制限付きサポート (シーケンスの 100,000 要素のみスキップ)	サポートなし
分布ジェネレーターとその他の BRNG 関連機能	<ul style="list-style-type: none"> 20 単変量 1 多変量 メソッド ID をパラメーターとして渡すことで複数の変換方法が利用可能 	<ul style="list-style-type: none"> 24 単変量 3 多変量 汎用離散ジェネレーター、連続単変量、データに基づく多変量ジェネレーター ランダム直交行列と相関行列、二元表、順序統計量、サンプルと置換、確率過程 個別のサブルーチンで複数の変換方法が利用可能 ORNG は一様分布のみ 	<ul style="list-style-type: none"> 24 単変量 2 多変量 汎用単変量分布ジェネレーター ランダムサンプル、置換、時系列モデル、直交行列と相関行列、乱数表 複数の変換方法は利用できない ORNG は一様、ガウス、対数正規分布向け
ベクトル/スカラー・インターフェイス	ベクトル・インターフェイスのみ	一部の分布ジェネレーターにはベクトル形式とスカラー形式の両方、その他はベクトル形式かスカラー形式のいずれかのみ	一部の分布ジェネレーターにはベクトル形式とスカラー形式の両方、その他はベクトル形式かスカラー形式のいずれかのみ
プログラミング言語インターフェイス	C と Fortran インターフェイスが 1 つのパッケージ	Fortran、C、Java* インターフェイスが個別のパッケージ	C と Fortran インターフェイスが個別のパッケージ
対応ハードウェア	インテル® アーキテクチャーのみ ターゲット・アーキテクチャー向けに高度に最適化	マルチプラットフォーム (例: インテル、Cray、HP、IBM、Sun)	マルチプラットフォーム (例: Apple、Cray、HP、IBM、Sun)

Mark 20 よりライブラリーで 2 通りの正規分布に従う乱数の生成方法を利用できます。

インテル® マス・カーネル・ライブラリー (インテル® MKL) 10.2: 詳細

次の表は、乱数生成機能を持つ 3 つのライブラリー、インテル® MKL 10.2、IMSL F90MP 6.0、NAG SMP Fortran 77 ライブラリー (Mark 21) の数値結果とパフォーマンス結果を示したものです (インテル® Xeon® プロセッサー・ベースのシステムを使用)。

速度向上は、最も遅いものと比較した倍率です。

性能比較: ブラックショールズのオプション価格モデル^{1,8}

ライブラリー	基本 ジェネレーター	オプション値 (厳密値)		絶対エラー (標準エラー)		時間 (秒)	速度向上 (倍)
		Call	Put	Call	Put		
インテル® MKL	MCG31m1	16.7306 (16.7341)	7.2177 (7.2179)	0.0036 (0.0019)	0.0002 (0.0009)	4.671	8.78
	MCG59	16.7364 (16.7341)	7.2162 (7.2179)	0.0023 (0.0019)	0.0017 (0.0009)	4.86	8.44
	MT19937	16.7349 (16.7341)	7.2164 (7.2179)	0.0007 (0.0019)	0.0015 (0.0009)	5.078	8.08
NAG	オリジナル	16.7339 (16.7341)	7.2182 (7.2179)	0.0002 (0.0019)	0.0003 (0.0009)	11.45	3.58
IMSL	MT19937	16.7324 (16.7341)	7.2178 (7.2179)	0.0017 (0.0019)	0.0001 (0.0009)	35.703	1.15
	最小基準	16.7343 (16.7341)	7.217 (7.2179)	0.0001 (0.0019)	0.0009 (0.0009)	41.031	1.00

1. モンテカルロは、複雑な世界市場の財務分析をリアルタイムで行う強力なツールです。ここでは、よく知られているブラックショールズのオプション価格モデルについて検討します。ブラックショールズのオプション価格モデルはオプション価格について考察する枠組みで、金融業界では事実上の業界標準となっています。

この評価では、IMSL ライブラリーの「最小基準」および 32 ビット MT19937 基本ジェネレーターと、NAG ライブラリーの「オリジナル」基本ジェネレーターを使用しています。インテル® MKL 10.0 では、MCG31m1、MT19937、MCG59 の 3 つの基本ジェネレーターを使用しています。MCG59 は、NAG ライブラリーの「オリジナル」基本ジェネレーターと同一で、MCG31m1 のプロパティは IMSL の「最小基準」基本ジェネレーターと同様です。

本性能評価についての詳細は、ホワイトペーパー『Making the Monte Carlo Approach Even Easier and Faster』(Sergey A. Maidanov, Andrey Naraikin 著) を参照してください。

<http://software.intel.com/en-us/articles/making-the-monte-carlo-approach-even-easier-and-faster/index.htm> (英語)

要約 - インテル® MKL のベクトル統計ライブラリーには、畳み込み/相関とインテル® アーキテクチャー向けに高度に最適化された、包括的な乱数ジェネレーター・セットが含まれており、ほかのソリューションよりも大幅にアプリケーションのパフォーマンスを向上させることができます。

資料

インテル® VSL パフォーマンス・データ: インテル® Xeon® プロセッサーおよびインテル® Itanium® プロセッサーにおけるインテル® VSL 乱数ジェネレーターと乱数分布についての詳細なパフォーマンス情報は、
http://software.intel.com/sites/products/documentation/hpc/mkl/vsl/vsl_data/vsl_performance_data.htm (英語) を参照してください。

インテル® VSL ノート (PDF 1.2MB): VSL BRNG と分布の詳細情報および乱数ジェネレーター全般に関しては、
<http://www.intel.com/cd/software/products/asm-na/eng/347649.htm> (英語) を参照してください。

Monte Carlo European Options Pricing (PDF 85KB): 金融派生商品価格における乱数ジェネレーターの使用についての論文は、
<http://software.intel.com/en-us/articles/monte-carlo-simulation-using-various-industry-library-solutions/> (英語) を参照してください。

LINPACK ベンチマーク

インテルでは、インテル® アーキテクチャー・ベース・システムで最高レベルのベンチマーク結果が得られる、インテル® MKL でビルドされた LINPACK ベンチマーク・パッケージを無料で提供しています。

Intel® Optimized LINPACK Benchmark パッケージを使用すると、インテル® アーキテクチャー・ベース・システムで最高レベルのベンチマーク結果を得るのに役立ちます。LINPACK ベンチマークを実装するこの無償パッケージは、インテル® Xeon® プロセッサ・ベースのシステムおよびインテル® Itanium® プロセッサ・ベースのシステムで最大限のパフォーマンスを引き出すように高度にチューニングされた BLAS、LAPACK ソフトウェアを使用します。これらと同じハイパフォーマンスな BLAS、LAPACK ルーチンがインテル® MKL で利用可能です。

インテルは、2つの LINPACK ベンチマーク・パッケージを提供しています。

Intel® Optimized SMP LINPACK Benchmark パッケージ

SMP マシン用。このパッケージには、LINPACK 1000 ベンチマーク・コード* (<http://www.netlib.org/benchmark/1000d>) の実装が含まれています。パフォーマンスおよびダウンロード情報は、下記を参照してください。

パッケージには、次にリストされている実行ファイル、shell スクリプトと入力ファイルが含まれています。

Intel® Optimized (SMP) LINPACK Benchmark 10.2

ファイル名	説明
linpack_itanium	Itanium® ベース・システム用 64 ビット版プログラム実行ファイル
linpack_xeon32	インテル® Xeon® プロセッサ・ベースのシステム用 32 ビット版プログラム実行ファイル
linpack_xeon64	64 ビット版インテル® Xeon® プロセッサ・ベースのシステム用 64 ビット版プログラム実行ファイル

Intel® Optimized MP LINPACK Benchmark for Clusters パッケージ

このパッケージは、分散メモリー型コンピューター・システム (クラスター) 上で使用する Massively Parallel MP LINPACK* (<http://www.top500.org/>) ベンチマークの実装です。Top 500 スーパーコンピューター・リストへの申告には、是非このパッケージを使用して、コンピューター・クラスターを評価してください。

Intel® Optimized MP LINPACK Benchmark 10.2 for Clusters

ファイル名	説明
HPL 1.0a	完全な HPL 1.0a ディストリビューションと HPL をより簡単に使用するための追加ファイル。
nodeperf.c	全クラスターノードで DGEMM パフォーマンスの一貫性をテストするためのプログラム。

使いやすさ

Intel® Optimized LINPACK Benchmark パッケージは、時間の節約を目的に設計されています。ファイルのダウンロードも少なく、コンパイルも不要、繰り返し作業はより少ないため、貴重な時間を節約します。Intel® Optimized SMP LINPACK Benchmark パッケージは、HPL のような時間のかかる検索作業は必要ありません。Intel® Optimized MP LINPACK Benchmark パッケージが検索を行うときは、プロセスの初期段階で情報を提供するため、再設定や別の実行を行うのに、実行全体が完了するまで待つ必要がありません。Intel® Optimized LINPACK Benchmark パッケージを試して、どの程度の節約ができるかをお試しください。

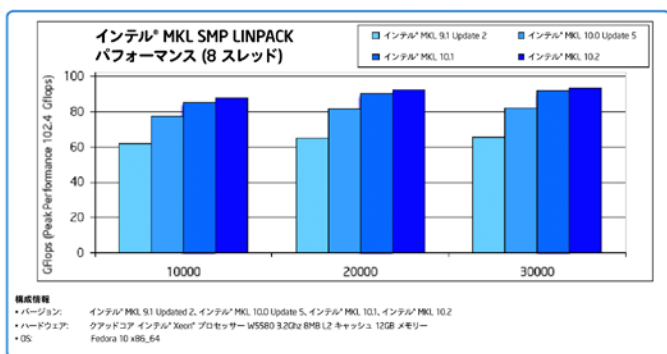
詳細は、インテル® MKL ユーザーズガイドの LINPACK ベンチマークの章を参照してください

経過時間 h:mm	Intel® Optimized SMP LINPACK バイナリー	経過時間 h:mm	HPL
0:00	開始	0:00	開始
0:00	Intel® Optimized LINPACK パッケージのダウンロード	0:00	HPL のダウンロード
0:10	スクリプトの設定	0:10	BLAS のダウンロード
0:20	実行の開始	0:20	MPI のダウンロード
1:00	結果の収集	0:30	MPI のビルド
		1:00	HPL のビルド
		1:30	実行 1 の開始
		2:30	結果のレビュー
		2:50	HPL.dat の変更
		3:00	実行 2 の開始
		4:00	結果のレビュー
			満足する結果が得られるまで HPL.dat の変更を繰り返す。延々と作業を繰り返すこともあります。
		8:00	今日はひとまずこれで終わりにして、明日は完了できるよう祈るのみ。

パフォーマンス

次のグラフは、Intel® Optimized SMP LINPACK Benchmark パッケージが達成した目覚ましいパフォーマンスを示しています。

- クアッドコア インテル® Xeon® プロセッサ・ベース・システムで理論上最大のパフォーマンスの 91.5 パーセントを達成。
- 新しいバージョンのインテル® MKL でパフォーマンスが向上。9.1 と比較すると、10.2 では最大 43% のパフォーマンスが向上。



ダウンロード

下記の Optimized LINPACK と MP LINPACK ベンチマーク・パッケージは、無料でダウンロードできます。

ダウンロード:

<http://www.intel.com/cd/software/products/asmo-na/eng/363191.htm> (英語)

パッケージ	ダウンロード・サイズ	パッケージの内容			
		Intel® Optimized LINPACK Benchmark		Intel® Optimized MP LINPACK Benchmark for Clusters	
		ソース	バイナリー	ソース	バイナリー
Linux® パッケージ (.tgz)	6.71MB		X	X	X
Windows パッケージ (.zip)	3.79MB		X	X	X
Mac OS パッケージ (.tgz)	763KB		X		

お問い合わせ/ご意見

LINPACK benchmark についてのフィードバックを、インテル® MKL ユーザーフォーラム

(<http://softwarecommunity.intel.com/isn/Community/en-US/forums/1273/ShowForum.aspx> (英語) まで是非お寄せください。

資料

インテル® クラスターツール: www.intel.co.jp/jp/software/products/

互換性

オペレーティング・システム

インテル® MKL 10.2 は、Linux、Windows (HPC Server 2008 を含む)、Mac OS X をサポートします。Linux 系の OS には、RedHat®, Suse、Debian、Ubuntu、Asianux® およびその他の Linux Standard Base 3.1 系が含まれます。

開発環境

インテル® MKL は、Microsoft Visual Studio®, Xcode®, Eclipse®, GNU* コンパイラ・コレクション (GCC) などの一般的なツールや開発環境と統合して使用することができます。

プロセッサ

インテル® MKL 10.2 は、すべてのインテル® アーキテクチャー互換プロセッサをサポートし、特に以下のプロセッサ向けに最適化されています。

- インテル® Xeon® プロセッサ・ファミリー
- インテル® Core™ プロセッサ・ファミリー
- インテル® Itanium® プロセッサ・ファミリー
- インテル® Pentium® プロセッサ・ファミリー
- AMD® Opteron® プロセッサ・ファミリーおよび Athlon® プロセッサ・ファミリー

注: インテル® MKL Mac OS X 版は単体製品としては提供されていません。インテル® C++ コンパイラ・プロフェッショナル・エディションとインテル® Fortran コンパイラ・プロフェッショナル・エディションにのみ付属しています。

テクニカルサポート

インテル® ソフトウェア開発製品をご購入いただくと、1 年間のサポートサービスを受けることができます。インテル® プレミアサポートでは、オンラインでユーザー登録をするだけで、インテル® MKL のディスカッション・フォーラム、テクニカルガイド、アプリケーション・ガイド、各種ドキュメントなどを入手できます。製品をインストールした後にユーザー登録を行うと、サポートならびに製品アップデートに関する情報をご利用いただけます。

インテル® プレミアサポート

インテル® MKL を購入いただくと、1 年間のサポートサービスを受けることができます。サポートサービスの有効期間内は、アップグレード・サービス (メジャーリリースを含む) を無料で受けることができます。詳細は、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 (<https://registrationcenter.intel.com/RegCenter/Register.aspx>) を参照してください。また、ユーザーフォーラムは、コミュニティー・サポートを得るための最適な場所です。

ユーザーフォーラム

インテル® MKL での体験を是非ほかのユーザーと共有してください。

インテルが管理するインテル® MKL ディスカッション・フォーラム:

<http://softwarecommunity.intel.com/isn/Community/en-US/forums/1273/ShowForum.aspx> (英語)

資料

インテル® MKL のドキュメントについては、

<http://www.intel.com/cd/software/products/asmo-na/eng/345631.htm>

(英語) を参照してください。

§ 性能に関するテストや評価は、特定のコンピューター・システム、コンポーネント、またはそれらを組み合わせて行ったものであり、このテストによるインテル製品の性能の概算の値を表しているものです。システムの設計、構成などの違いにより、実際の性能は掲載された性能テストや評価とは異なる場合があります。システムやコンポーネントの購入を検討される場合は、ほかの情報も参考にして、パフォーマンスを総合的に評価することをお勧めします。インテル製品の性能評価についてさらに詳しい情報をお知りになりたい場合は、http://www.intel.co.jp/performance/resources/benchmark_limitations.htm を参照してください。

© 2009 Intel Corporation. 無断での引用、転載を禁じます。Intel、インテル、Intel ロゴ、Intel Core、Pentium、Itanium、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。

JPN/0912/PDF/XL/SSG/TT 321515-001JA

