

インテル® Xeon Phi™ コプロセッサ向けのサポートを利用して R を実行する

この記事は、インテル® デベロッパー・ゾーンに公開されている「[Running R with Support for Intel® Xeon Phi™ Coprocessors](#)」の日本語参考訳です。

概要

R は、統計計算および解析向けの無料のオープンソース・ソフトウェア環境です。広範な UNIX* および Linux* プラットフォーム、Windows*、macOS* でビルドおよび実行できます。R プロジェクトの Web サイト <http://www.r-project.org/> (英語) からソースコードをダウンロードして実行ファイルをビルドするか、サポートされる各 OS 用の事前ビルド済み実行ファイルをダウンロードすることができます。ほとんどのユーザーは、事前ビルド済み実行ファイルをダウンロードします。この記事では、インテル® コンパイラーとインテル® マス・カーネル・ライブラリー (インテル® MKL) を使用して R をビルドし、ビルドした実行ファイルをインテル® Xeon® プロセッサおよびインテル® Xeon Phi™ コプロセッサ上で実行する方法を示します。

R とインテル® ソフトウェア・ツールの入手

R をダウンロードするには、次の操作を行います。

- <http://www.r-project.org/> (英語) に移動して、ページの左側にある "CRAN" (Comprehensive R Archive Network) リンクをクリックします。CRAN ミラーの一覧が表示されます。
- 使用するミラーのリンクをクリックします。
- 表示されるページの左側にある "R Sources" リンクをクリックします。
- 最新のソースコード・リリースは、ページの上部の目立つところに表示されているはずです。リンクをクリックして、ブラウザの指示に従ってソフトウェアをダウンロードします。
- パッケージは通常、アーカイブおよび圧縮されています。ご使用の OS 用の適切な展開ツールを使用して、ソフトウェアをインストールします。

インテル® ソフトウェア・ツールをダウンロードするには、次の操作を行います。

- <https://www.isus.jp/intel-parallel-studio-xe/> に移動します。
- ツールを評価するか、購入するかを決定します。購入する場合は、[価格] ボタンをクリックします。評価する場合は、[評価版] ボタンをクリックします。
- 表示されるページの指示に従って、評価または購入手続きを完了してください。
- ライセンスファイル、シリアル番号、インストール手順を含むメールが送られてきます。手順に従ってツールをダウンロードして、インストールしてください。

インテル® ソフトウェア・ツールのビルド

Linux* では、インテル® ソフトウェア・ツールがデフォルトの場所 (/opt/intel/composerxe) にインストールされている場合、次のコマンドを実行してビルドします。

```
$ source /opt/intel/composerxe/bin/compilervars.sh intel64
$ ./configure --with-blas="-L/opt/intel/composerxe/mkl/lib/intel64
-lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core -liomp5 -lpthread -lm" --with-
lapack CC=icc CFLAGS=-O2 CXX=icpc CXXFLAGS=-O2 F77=ifort FFLAGS=-O2 FC=ifort
FCFLAGS=-O2
$ make
$ make check
```

R のビルドに関する詳細は、次の記事を参照してください。

- <https://www.isus.jp/products/c-compilers/build-r-301-with-intel-c-compiler-and-intel-mkl-on-linux/>
- <https://www.isus.jp/products/mkl/using-mkl-with-r/>

インテル® MKL を利用して R をビルドすると、最適化された行列乗算ルーチンを多くの R データ解析計算で利用することができます。

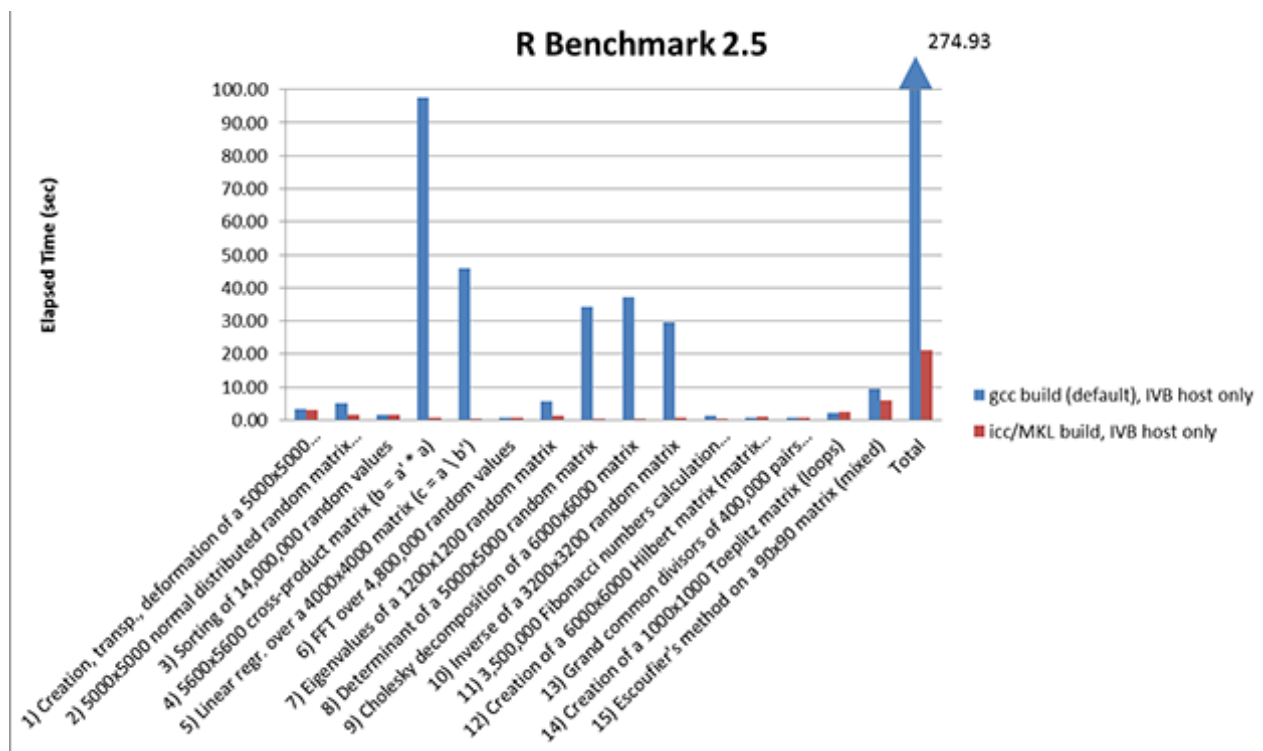
R 実行ファイルをビルドしたら、デフォルトのビルドやダウンロードした実行ファイルと同様に実行します。

インテル® Xeon® プロセッサ上でのベースライン・パフォーマンス

各種 Linux* 向けの事前ビルド済み実行ファイルは、GNU* ツールを利用してビルドされています。残念ながら、これらの実行ファイルは、マルチコアシステム上でも、並列に実行可能な行列演算でシングルスレッド・パフォーマンスしか達成できません。以下の表は、Red Hat* Enterprise Linux* 6.3 上で、インテル® C++ コンパイラー (icc) 14.0.1 とインテル® MKL でビルドした R と gcc 4.4.6 でビルドされた "デフォルト" のパフォーマンスを比較したものです。インテル® MKL でビルドした実行ファイルは、行列演算をマルチコアで実行するため、これらの演算を非常に高速に実行できます。テストには、以下の Web サイトにある R benchmark-2.5 を使用しました。

<http://r.research.att.com/benchmarks/R-benchmark-25.R> (英語)

大きなワークロード・サイズをテストするため行列サイズを増やしています。テストの結果、インテル® MKL でビルドされた R は、gcc ビルドよりも 13 倍も高速になりました。



テスト	gccビルドの実行時間 (秒)	icc/インテル® MKLビルドの実行時間 (秒)
5000x5000 行列の作成、転置、変形	3.25	2.95
5000x5000 正規分布乱数行列 ^1000	5.13	1.52
14,000,000 乱数値の並べ替え	1.61	1.64
5600x5600 クロス積行列 (b = a' * a)	97.44	0.56
4000x4000 行列の線形回帰 (c = a \ b')	46.06	0.49
4,800,000 乱数値の FFT	0.65	0.61
1200x1200 乱数行列の固有値	5.55	1.37
5000x5000 乱数行列式	34.18	0.55
6000x6000 行列のコレスキー分解	37.07	0.47
3200x3200 乱数行列の逆数	29.49	0.57
3,500,000 フィボナッチ数列の計算 (ベクトル演算)	1.31	0.38
6000x6000 ヒルベルト行列の作成 (行列演算)	0.77	0.99
400,000 ペアの最大公約数 (再帰)	0.63	0.56
1000x1000 テプリッツ行列 (ループ)	2.24	2.34
90x90 行列に対するエスコフィエの図式解放 (混在)	9.55	6.02
合計	274.93	21.01

システム構成:

- 2 ソケット/24 コア
- プロセッサ: インテル® Xeon® プロセッサ E5-2697 v2 @ 2.70GHz (12 コア)、インテル® ハイパースレッディング・テクノロジー有効
- オペレーティング・システム: Red Hat® Enterprise Linux® 2.6.32-358.6.2.el6.x86_64.crt1 #4 SMP Fri May 17 15:33:33 MDT 2013 x86_64 x86_64 x86_64 GNU*/Linux*
- メモリー: 64GB
- コプロセッサ: 2x インテル® Xeon Phi™ コプロセッサ 7120P: 61 コア @ 1.238GHz、4-way インテル® ハイパースレッディング・テクノロジー有効、メモリー: 15872MB
- インテル® メニーコア・プラットフォーム・ソフトウェア・スタック 2.1.6720-16
- インテル® C++ コンパイラ 13.1.3 20130607 (2013.5.192)

インターネットを検索すると、独立した、同様の比較結果が得られます。Revolution Analytics の記事 (<http://www.revolutionanalytics.com/revolution-revor-enterprise-benchmark-details> (英語)) では、インテル® コンパイラでビルドされた独自の R 製品を Windows* 上でテストした結果を説明しています。<http://www.r-bloggers.com/speeding-up-r-with-intels-math-kernel-library-mkl/> (英語) には、汎用 BLAS でビルドされた R とインテル® MKL でビルドされた R を Ubuntu* 上で実行した結果の比較があります。

インテル® Xeon Phi™ コプロセッサ向けの R サポート

インテル® ソフトウェア・ツールを利用してビルドするその他の利点は、インテル® Xeon Phi™ コプロセッサが搭載されているシステムでは、インテル® MKL が特定の並列行列演算をコプロセッサへ自動でオフロードできることです。インテル® ソフトウェア・ツール向けの前述の手順に従ってビルドした場合、R をインテル® Xeon Phi™ コプロセッサで実行するための準備はすでに完了しています。次の環境変数を設定して、インテル® MKL に行列演算をオフロードするように指示できます。

```
$ export MKL_MIC_ENABLE=1
```

R が行列演算を開始すると、インテル® MKL はプラットフォームのホスト・プロセッサとコプロセッサ間でワークを分割します。MKL_HOST_WORKDIVISION および MKL_MIC_0_WORKDIVISION 環境変数でワークの比率を指定できます。例えば、次のように指定します。

```
$ export MKL_HOST_WORKDIVISION=0.1  
$ export MKL_MIC_0_WORKDIVISION=0.9
```

この場合、インテル® MKL はワークの 90% をインテル® Xeon Phi™ コプロセッサに送り、10% をホスト・プロセッサで実行します。インテル® Xeon Phi™ コプロセッサが 2 つある場合は、次のように指定します。

```
$ export MKL_HOST_WORKDIVISION=0.2  
$ export MKL_MIC_0_WORKDIVISION=0.4  
$ export MKL_MIC_1_WORKDIVISION=0.4
```

この場合、インテル® MKL はワークの 80% をインテル® Xeon Phi™ コプロセッサに送り (2 つのカード間で等分して 40% ずつ実行)、20% をホスト・プロセッサで実行します。さまざまな比率を試して、最適なワークの比率を見つけることができます。

インテル® MKL の自動オフロードのパフォーマンスは、R で解析するワークロードのサイズに大きく依存します。インテル® MKL には、自動オフロードにより利点を得るのに十分なワークロード・サイズを決定するためのヒューリスティックのセットが含まれています。これは、引き続き調査中の分野です。ぜひ皆さんのワークロードで試してみて、結果を共有してください。

まとめ

インテル® ソフトウェア・ツールでビルドした R は、GNU* ツールで事前ビルドされたまたはユーザーによってビルドされた実行ファイルと比較して、パフォーマンスが大幅に向上しました。この記事では、R とインテル® ソフトウェア・ツールをダウンロードしてビルドし、インテル® MKL の自動オフロード機能によりホスト・プロセッサとインテル® Xeon Phi™ コプロセッサで R を実行する方法を説明しました。実際にこの方法を試してみて、その結果を共有してください。

コンパイラーの最適化に関する詳細は、[最適化に関する注意事項](#)を参照してください。