

無償BLAS/LAPACKライブラリによる 浮動小数点演算ベンチマーク結果

ベンチマークを実施した切っ掛け

- (1) Core 2 Quadマシンの調子がおかしくなったので、2015年に発売された第6世代のIntel CPU(Skylake)マシンを購入した
- (2) Intel Core i シリーズは、拡張命令が進化している。Wikipediaによると従来のSSE3,4と比べて、AVX(Sandy Bridge)で2倍、AVX2 (Haswell)で4倍の演算能力[1]
- (3) Scilab用ベンチマークスクリプト／行列演算のFLOPS計算が公開されているので[2]、理論値と比較が可能
- (4) Rのベンチマーク結果が多数公開されているので、相対比較可能
- (5) Windows版のScilabは、古いバージョン?のIntel MKLがデフォルトでインストールされ、理論値に近い値が得られた。併せて、Linux/OpenBlasの環境で比較した。

WindowsとLinuxでのベンチマーク結果の概要 (Scilab、R)

Windows用Scilabはデフォルト環境で高速。LinuxでOpenBlasをインストールすると、理論値に近い演算能力が得られた (Skylakeではソースからのビルドが必要)。

1. ベンチマークを行ったマシン(全て4コア、OSは64bit版)

型番	CPU	RAM	OS	Disk構成
HPE-190JP	Core i7 960 3.2GHz	12GB	デュアルブート ▪ Windows 7 Pro ▪ Debian 8	▪ 2TB / HDD ▪ 320GB / HDD
HP EliteBook 8460w	Core i7 2630QM 2GHz	8GB	内蔵とeSATAのデュアルブート ▪ Debian 8 ▪ Arch Linux、Debian-Sid (コンソールモード)	▪ 500GB / HDD (Windows 7をDebian 8-KDEに入れ替え) ▪ 2TB / eSATA接続HDD (PCオーディオ用、スタンドアローン)
HP ENVY 17	Core i7 4700MQ 2.4GHz	16GB	Window 8.1 Pro	▪ 256GB*2 / SSD
自作マザーボード ASUSTek H170-Pro	Core i5 6600K 3.5GHz	32GB	デュアルブート ▪ Windows 7 Pro ▪ Ubuntu 15.10	▪ 480GB / SSD + 3TB / HDD ▪ 1TB / HDD

Linuxの入れ替えが容易なように、基本的に1ドライブに1OSインストール

独り言 —趣味のオーディオについて—

1. メインのオーディオ : USB-DDコンバータによりS/PDIFに変換して、古いDAコンバータ (Birdland Audio / U.S.AのODEON-LITE)からオーディオ装置に接続
2. PC用のサウンドカード / Onkyo SE-200PCI ltdは、当時PC用としては高音質といわれていたがODEON-LITEと比べると、馬力が足りないなど圧倒的な差がある。安物の中華デジタルアンプと小型スピーカー(サブオーディオ)でもすぐに分かる(音源はwaveファイル)。
3. SE-200PCI ltdにおいて、別のLinux PCのマザーボードから光入力して音を出した方がWindows 7よりも音が良かった。なお、SE-200PCI ltdはオンボード音源よりは高音質だと考え、PCIスロットのあるマザーボードを選択した。PC電源は高グレードとした。
4. メインのオーディオ用Linuxは、コンソールモードでミニマムインストールとして、ファイル操作とカーネルのビルド・ALSA設定用にXfce4をインストール。Display Managerをインストールしていないので、startxfce4(arch), startx(debian)を入力しないとXが起動しない。ソフト(music on console)からALSAへ直接出力するように設定。カーネルのパッチ適用・config設定・ビルド、ALSA設定、静音化のパワー設定などが面倒。Xなしの方が音がクリア。
5. Linux-Kernelのバージョンアップにより音質が向上した。
3.2では粗さが目立ったが、3.14でオーディオ用CDプレイヤー並みの音質になった。
なお、DSDレコーディングのSACD(アコースティックのジャズ, クラシック)には勝てません。
6. Arch Linux(切れが良い)とDebian(奥行のある一オーケストラ向き)で音像が異なる。

マザーボード(ASUS H170Pro): 複数ドライブ環境のデュアルブート構築

- Installing Ubuntu in UEFI mode[10]によると、レガシーBIOS/GPT-UEFIを合わせるとの記載があった。
- SATA接続のドライブにマザーボードのDVD、USB接続のDVDからWindows 7をUEFIモードでブートし、Windowsをインストールした。
- Windowsのドライブを取り外して、別のHDDにUbuntuをUEFIモードでインストールした。
- この状態で作成したシステム修復ディスクを試しに起動すると、正常に起動しなくなった。Linuxのドライブを外して、再起動すると自動修復できたが、PC1台で2個のEFIシステムパーティション(ESP)を使うのは危険なことを痛感した。



- マザーボード(ASUS H170Pro)でドライブを分ける環境において、以下の方法で順調に動作している。なお、Linuxの選択は、起動時にF8を押して、Linuxをインストールしたドライブを選ぶ。
- Windows 7は、UEFIモードでインストール(グラフィックボードELSAのホームページにおいてUEFI BIOSが必要と記載があったのでUEFIモードにした)
 - Linuxは、レガシーBIOSモードでインストール(2.2TBの壁は気にしない、データ用はUSB3.0接続であれば高速、内蔵ドライブ追加も可)
 - Windows 10にアップグレードしない方が吉?

2. 使用したベンチマークスクリプト、ライブラリ

BLAS/LAPACKのライブラリの変更で高速化が可能なscilabとRにより比較した

1. Scilab用bench_matmul.sce [2]

- $n \times n$ の乱数行列の計算時間からMFLOPSを求めるスクリプト。
- MFLOPSで出力されるので、CPUの理想的な性能との比較が可能

2. R用R-benchmark-25.R [3]

- 総合的なベンチマーク、時間で比較
- ベンチマーク結果が多い(「OpenBlas R 高速化」、「MKL R 高速化」で検索するとすぐに見つかる)

Windows版 Intel Math Kernel Library 10.3 for Scilab 5.5.x

C:\Program Files\scilab-5.5.2\bin\readme-mkl.txtの抜粋 (LicenseやRelease Notesでは11.0)

Scilab 5.5.x uses Basic Linear Algebra Subprograms BLAS

Scilab also uses BLAS behind its core numerical linear algebra routines from LAPACK.

Processors Minimum Recommended Supported :

* Intel Core processor family (世代不明のXeon、Pentium4、AMD社CPUは省略)

<http://www.intel.com/software/products/mkl>

Intel Software Development Products License Agreement :

<http://software.intel.com/en-us/articles/end-user-license-agreement>

Intel Math Kernel Library 11.0 Licensing FAQ :

<http://software.intel.com/en-us/articles/intel-math-kernel-library-licensing-faq/>

Intel Math Kernel Library 11.0 for Windows* - Release Notes

<http://software.intel.com/en-us/articles/intel-mkl-110-release-notes>

Allan CORNET - Consortium Scilab - DIGITEO 2010-2011

Antoine ELIAS - Scilab Enterprises - 2013

Ubuntu 15.10版OpenBlas (Synapticのコメントより)

Optimized BLAS (linear algebra) library (shared library)

OpenBLAS is an optimized BLAS library based on GotoBLAS2 1.13 BSD version.

Unlike Atlas, OpenBLAS provides a multiple architecture library. All kernel will be included in the library and dynamically switched to the best architecture at run time (only on x86 arches).

追記: Skylakeマシンでは、自動検出に失敗した。

HASWELLの仮想マシンでは、64bit対応のx86全cpuのビルドするように動作したので、HASWELL指定でビルドした。その後で、リポジトリのバイナリで同等性能であること確認した。

For more information on how to rebuild locally OpenBLAS, see the section:"Building Optimized OpenBLAS Packages on your ARCH" in README.Debian

1. OpenBlas-0.2.14のintel製CPUの対応 - TargetListより

P2, KATMAI, COPPERMINE, NORTHWOOD, PRESCOTT, BANIAS,
YONAH, CORE2, PENRYN, DUNNINGTON, **NEHALEM,**
SANDYBRIDGE, HASWELL, ATOM

2. ソースコードからのビルド方法 [4]

debファイルを作成し、自動的にblas/lapackとリンクさせるため、
Ubuntu15.10のソースからビルドした。

```
$ apt-get source openblas
```

```
$ sudo apt-get build-dep openblas
```

展開されたフォルダに移動

```
$ dpkg-buildpackage -b -uc... (CPUの自動検出に失敗・停止)
```

debian/rules を編集

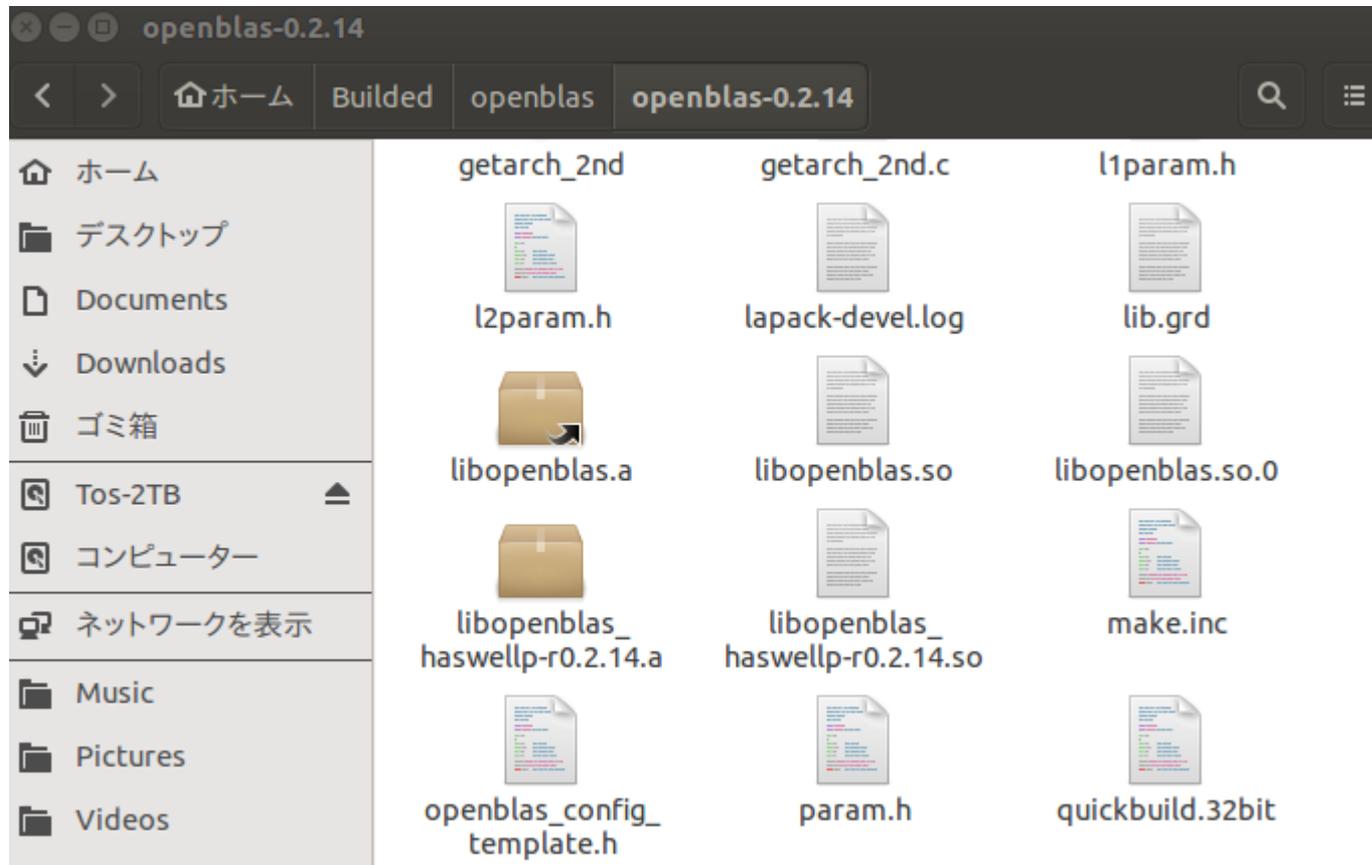
ENABLE_DYNAMIC_ARCHS をコメントアウトし、

GENERIC_OPTIONS += TARGET=HASWELL によりCPU指定

```
$ dpkg-buildpackage -b -uc を再実行
```

Core i5 6600K 用debパッケージのビルド結果

展開したフォルダ内: Haswell用のライブラリが作成された

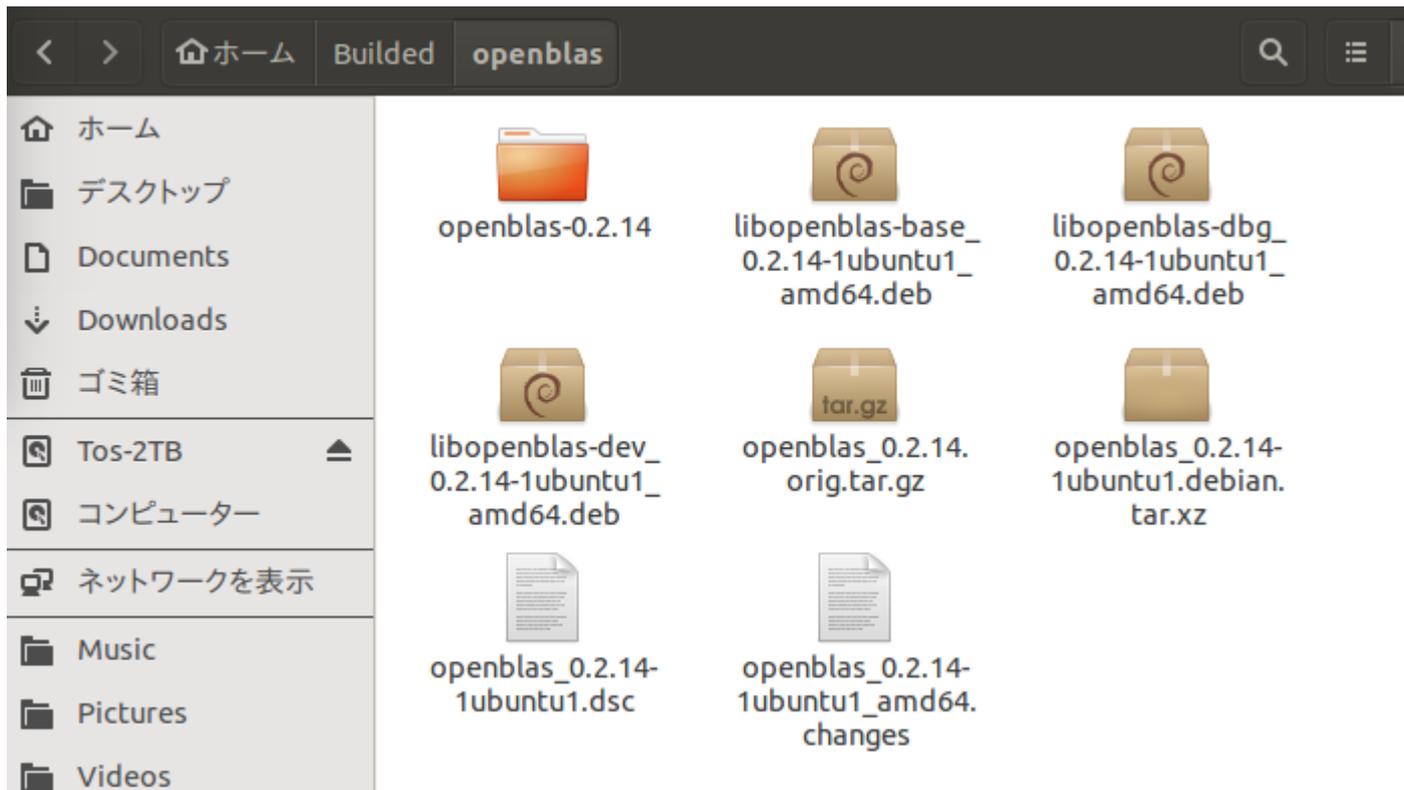


Core i5 6600K 用debパッケージのビルド結果(2)

一つ上のフォルダに3個のdebファイルが作成された。
インストールは、以下のコマンドを実行

```
$ cd ..
```

```
$ sudo dpkg -i *.deb
```



実行時の使用ライブラリの確認－参考[5]

Scilab、R共にopenblasのライブラリを使用していることを確認

Scilab実行時のライブラリ (lsof -p 2353 | grep 'blas¥|lapack')

/usr/lib/openblas-base/liblapack.so.3

/usr/lib/libopenblas_haswellp-r0.2.14.so

/usr/lib/openblas-base/libblas.so.3

R実行時のライブラリ (lsof -p 2212 | grep 'blas¥|lapack')

/usr/lib/openblas-base/liblapack.so.3

/usr/lib/libopenblas_haswellp-r0.2.14.so

/usr/lib/openblas-base/libblas.so.3

注記: Linuxの端末では、¥はバックスラッシュ

3. ベンチマーク測定 Scilab / bench_matmul.sce

CPUは、全て4コア。 OSは、64bit版

CPU	OS	Scilabのバージョン	ライブラリ	速度:GFOPS (理論値)
Core i7 960 3.2GHz	Debian 8 (GNOME)	5.5.1	OpenBlas 0.2.12 リポジトリ	49.8 (51.2)
	Windows 7 Pro	5.5.2	MKL 10.3? for Scilab	45.2 (51.2)
Core i7 2630QM 2.0GHz	Debian 8 (KDE*)	5.5.1	OpenBlas 0.2.12 リポジトリ	59.8 (64)
Core i7 4700MQ 2.4GHz	Windows 8.1 Pro	5.5.2	MKL 10.3? for Scilab	124 (154)
Core i5 6600K 3.5GHz	Ubuntu 15.10	5.5.2	OpenBlas 0.2.14 リポジトリ	44 (224?)
			OpenBlas 0.2.14 ビルド	179 (224?)
	Windows 7 Pro	5.5.2	MKL 10.3? for Scilab	167 (224?)

*AMD(旧ATI)のcatalystドライバがGNOME非対応のため、フルデスクトップ機能のKDEとした

R / R-benchmark-25.R

	Core i7 3770K 3.5GHz 4コア RAM 16GB Ubuntu 13.10 [6]		Core i5 6600K 3.5GHz 4コア RAM 32GB	
			Ubuntu 15.10	Windows 7 Pro
ライブラリ	GNU	インテル® MKL & インテル® コ ンパイラー	OpenBlas-0.2.14 ソースからビルド	デフォルト環境
2800x2800 行列のク ロス積 ($b = a' * a$)	9.4276667	0.2966667	0.1556667	12.08667
3000x3000 行列のコ レスキー分解	3.7346666	0.1713333	0.1126667	4.783333
15 種類の全テスト の合計時間	29.1816667	5.4736667	4.0253333	30.53

単位: 秒

第6世代のCore i シリーズでは、Linux環境でHaswell指定でソースからopenblasをビルドすれば、行列演算が数十倍速くなる

注記: MKLの資料は第3世代のCore i 7

なお、最新のOpenBlas-0.2.15の単独ビルドでは、Haswellとして自動認識された

仮想マシン上でのベンチマーク

仮想ソフトウェア: Vmware Workstation 12 Player

マシン	Core i5 6600K 3.5GHz, RAM 32GB	Core i7 4700MQ 2.4GHz, RAM 16GB	
ホストOS	Windows 7 Pro	Windows 8.1 Pro	
ゲストOS	Ubuntu 15.10 2コア割り当て	Ubuntu 15.10 3コア割り当て	
ライブラリ	Openblas-0.2.14をソースからビルド (HASWELL)	Openblas-0.2.14 リポジトリ、ビルド 同等の速度	
Scilab bench_matmul.sce	97.4 GFLOPS	92.5 ~ 109 GFLOPS 温度によるターボ・ブーストの差?	
R 秒	2800x2800 行列のクロス積 ($b = a' * a$)	0.3043333	0.307
	3000x3000 行列のコレスキー分解	0.1963333	0.19933333
	15 種類の全テストの合計時間	5.069	5.9859999

仮想マシン上においても、割り当てコア数相当の速度が出た

参考情報 Intel MKL 無償に

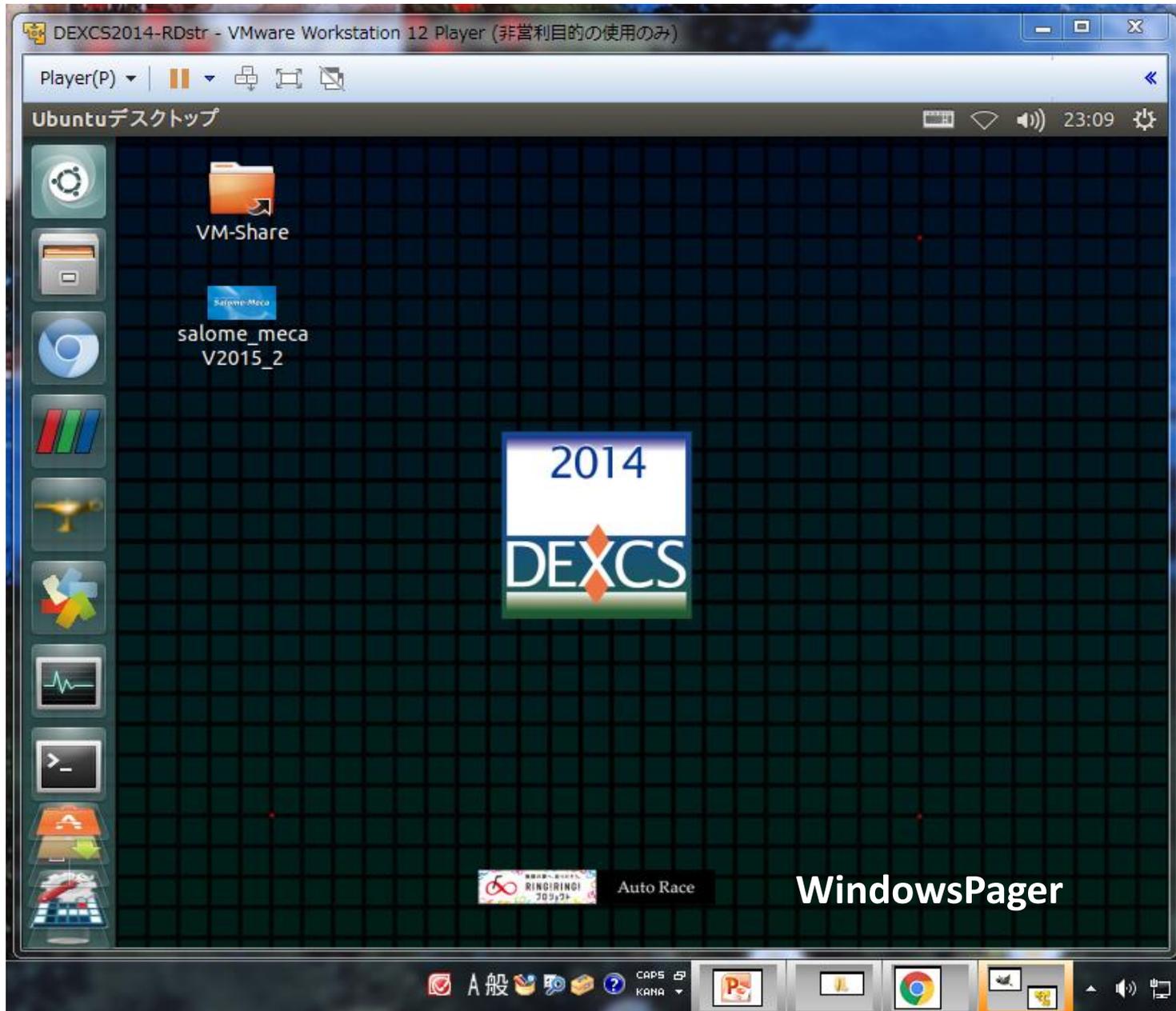
No Cost Options for Intel Math Kernel Library (MKL), Support Yourself, Royalty –Free -Submitted by James Reinders (Intel) on August 31, 2015
https://software.intel.com/en-us/articles/free_mkl

以下のバージョン／ライセンスがある(ユーザー登録が必要)

- Community Licenses for Everyone: Community Licensing for Intel Performance Libraries
- Evaluation Copies for Everyone: Try before you buy
- Use as an Academic Researcher: Qualify for Use as an Academic Researcher
- Student, Teacher: Qualify for Use as a Student, Educator
- Use as an Open Source Contributor: Qualify for Use as an Open Source Contributor

注記: デフォルトで/opt/intelにインストールされる。コンパイラのライブラリとのリンクなど、利用するにはスキルが必要と思われる[7]。

おまけ Windows 7 用仮想デスクトップソフト [8] [9]



まとめ

1. Windowsのデフォルト環境

ScilabはMKLがインストールされ高速。Rは遅い。

2. Linux環境

- (1) 第2世代:リポジトリのopenblas適用により単純な行列演算では初代Core i7マシン(SSE3/SSE4)の約2倍
- (2) 第4世代、第6世代:openblas(Haswell)適用により3~4倍の速度になった。
→(1), (2) よりcore i7 960のマシンは、技術計算用としては引退?
- (3) 高速化はvmware playerの仮想環境においても有効

3. 希望事項

次のUbuntu 16.04は、長期サポート版になると思われます。
次回のDEXCS作成は、できればHaswellに最適化したライブラリの追加をお願いしたい。
ライセンス上の問題がなければ、Intel MKLも検討してほしい。

参考資料ーその1

1. FLOPS

<https://ja.wikipedia.org/wiki/FLOPS>

2. Scilabのベンチマークスクリプト bench_matmul.sce

http://forge.scilab.org/index.php/p/docprogscilab/source/tree/HEAD/en_US/scripts/bench_matmul.sce

3. Rのベンチマークスクリプト R-benchmark-25.R

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

4. DebianやUbuntuで公式パッケージのソースをダウンロード／ビルドする

<http://note.kurodigi.com/debian-srcpkg/>

<https://www.debian.org/doc/manuals/apt-howto/ch-sourcehandling.ja.html>

5. OpenBLAS で R を高速化

http://qiita.com/hoxo_m/items/aa04b9d3d7a32cb6a1a3

6. オープンソース Python*、R、Juliaベースの HPC アプリケーションの高速化

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)によるパフォーマンスの向上

http://www.isus.jp/wp-content/uploads/pdf/PU17_opensource.pdf

7. mkl numpyのインストール方法

<http://qiita.com/unnonouno/items/8ab453a1868d77a93679>

参考資料ーその2

8. 仮想的な4つのデスクトップ画面を使って楽に作業ができるフリーソフト「WindowsPager」 <http://gigazine.net/news/20120722-windowspager/>
解凍すると実行ファイルが作成される。レジストリを操作しません。
64bit版のwindowspager.exeをダブルクリックするだけで動作する。
9. WindowsPagerのダウンロード先
<http://windowspager.sourceforge.net/>
10. UEFI ubuntu documentation
<https://help.ubuntu.com/community/UEFI>