# How to Improve R Computation using a Customized BLAS

Erin Melcon, Christopher Aden

May 8, 2012

# Why Use a Customized BLAS?



**Definition (BLAS)**

The Basic Linear Algebra Subprograms (BLAS) are a set of libraries, usually written in a low-level language like Fortran or C, to perform linear algebra routines (transpose, dot products, matrix inverses, etc).

R ships with a generalized BLAS designed to work with all CPU architectures.

Does not take advantage of CPU-specific optimizations.

Replacing it with a version specific to your architecture results in highly optimized numerical linear algebra.

# Setup on Windows

- Disable CPU Power Management
- Determine what processor model you have
- Download a pre-compiled dynamic version of Goto-BLAS from the website:

  http://prs.ism.ac.jp/ nakama/SurviveGotoBLAS2/binary/windows/
- Drop it in Program Files/R/bin/x64 (overwrite your current Rblas.dll)

# The Speed-Up



| Windows | Before | After |
|---|---|---|
| Creation, transp., deformation | **0.3766** | 0.6633 |
| Normal random matrix times 1000 | 0.9033 | **0.9000** |
| Sorting of random values | 0.8400 | **0.8266** |
| Cross-product pf a matrix | 14.483 | **0.6766** |
| Linear regression | 6.7133 | **0.5799** |
| Fast Fourier Transform | **0.6966** | 0.6999 |
| Eigenvalues of a random matrix | **1.1133** | 14.136 |
| Determinant of a random matrix | 4.8500 | **1.0000** |
| Cholesky decomposition of a matrix | 5.0833 | **0.8233** |
| Inverse of a random matrix | 4.6866 | **1.1899** |
| Fibonacci numbers (vector calc) | **1.4566** | 1.4766 |
| Hilbert matrix (matrix calc) | 0.5666 | **0.5499** |
| Grand common divisors (recursion) | **0.9433** | 1.0400 |
| Toeplitz matrix creation(loops) | **0.8899** | 0.9233 |
| Escoufier's method (mixed) | **0.5299** | 0.6699 |
| Overall Time | 44.1333 | **26.1566** |
| Trimmed Geometric Mean Time | 1.5799 | **0.8509** |

Over all the matrix computations, the updated BLAS resulted in a 1.68 times speed-up.

# Setup on Ubuntu/Debian

- Install R from a CRAN mirror with `sudo apt-get install r-base`
- Download and install atlas with `sudo apt-get install libatlas3gf-base`
- R automatically links ATLAS and makes it the default BLAS.

# The Speed-Up



| Ubuntu/Debian | Before | After |
|---|---|---|
| Creation, transp., deformation | 0.4393 | **0.365** |
| Normal random matrix times 1000 | 0.4306 | **0.3623** |
| Sorting of random values | 0.6320 | **0.59** |
| Cross-product pf a matrix | 10.753 | **0.4393** |
| Linear regression | 5.1506 | **0.3023** |
| Fast Fourier Transform | 0.3766 | **0.299** |
| Eigenvalues of a random matrix | 0.6790 | **0.5626** |
| Determinant of a random matrix | 2.6733 | **0.355** |
| Cholesky decomposition of a matrix | 3.9083 | **0.363** |
| Inverse of a random matrix | 2.5473 | **0.3723** |
| Fibonacci numbers (vector calc) | 0.6939 | **0.6403** |
| Hilbert matrix (matrix calc) | 0.3723 | **.2763** |
| Grand common divisors (recursion) | **1.6010** | 1.7076 |
| Toeplitz matrix creation(loops) | 0.6193 | **0.5460** |
| Escoufier's method (mixed) | 0.3739 | **0.3260** |
| Overall Time | 31.251 | **7.508** |
| Trimmed Geometric Mean Time | 1.0068 | **0.40872** |

Over all computations, the updated BLAS resulted in a 4 times speed-up.

# Disadvantages

Some limitations of the current method of updating BLAS:

- Does not take into account all cores. Thus, will not run at full performance.
- Not tuned to specific CPU architecture (but still better than default for most operations).

# Possible Solutions

It is possible to build `R` s.t. it will use the optimally tuned BLAS in most operating systems.

This is non-trivial (particularly in Windows), and a work in progress.

# Sources

- http://math-atlas.sourceforge.net/
- http://cran.r-project.org/doc/manuals/R-admin.html#BLAS
- http://cran.r-project.org/bin/windows/rw-FAQ.html#Can-I-use-a-fast-BLAS_003f
- http://prs.ism.ac.jp/~nakama/SurviveGotoBLAS2/binary/windows/
- http://r.research.att.com/benchmarks/