

Package ‘gcbd’

September 28, 2016

Type Package

Title 'GPU'/CPU Benchmarking in Debian-Based Systems

Version 0.2.6

Date 2016-09-27

Author Dirk Eddelbuettel

Maintainer Dirk Eddelbuettel <edd@debian.org>

Description 'GPU'/CPU Benchmarking on Debian-package based systems

This package benchmarks performance of a few standard linear algebra operations (such as a matrix product and QR, SVD and LU decompositions) across a number of different 'BLAS' libraries as well as a 'GPU' implementation. To do so, it takes advantage of the ability to 'plug and play' different 'BLAS' implementations easily on a Debian and/or Ubuntu system. The current version supports

- 'Reference BLAS' ('refblas') which are un-accelerated as a baseline
 - Atlas which are tuned but typically configure single-threaded
 - Atlas39 which are tuned and configured for multi-threaded mode
 - 'Goto Blas' which are accelerated and multi-threaded
 - 'Intel MKL' which is a commercial accelerated and multithreaded version.
- As for 'GPU' computing, we use the CRAN package
- 'gputools'

For 'Goto Blas', the 'gotoblas2-helper' script from the ISM in Tokyo can be used. For 'Intel MKL' we use the Revolution R packages from Ubuntu 9.10.

License GPL (>= 2)

LazyLoad yes

Depends R (>= 2.11.1)

Imports Matrix, DBI, RSQLite, plyr, reshape, lattice

Suggests gputools

SystemRequirements Debian or Ubuntu system with access to Goto Blas, Intel MKL, Atlas development build as well as a Nvidia GPU with CUDA support

OS_type unix

NeedsCompilation no

Repository CRAN

Date/Publication 2016-09-28 07:24:48

R topics documented:

analysis	2
benchmark	2
figures	3
utilities	4

Index	6
--------------	----------

analysis	<i>Analysis functions for GPU/CPU Benchmarking</i>
----------	--

Description

Analysis functions for GPU/CPU Benchmarking

Usage

```
loglogAnalysis()
```

Details

loglogAnalysis retrieves past benchmark results from the database contained in the package and returns intercepts and slopes of regressions of elapsed times on matrix dimensions (where both inputs are in logarithms).

benchmark	<i>Benchmarking functions for GPU/CPU Benchmarking</i>
-----------	--

Description

Benchmarking functions for GPU/CPU Benchmarking

Usage

```
getMatrix(N)
matmultBenchmark(N, n, trim=0.1)
matmultBenchmarkgputools(N, n, trim=0.1)
qrBenchmark(N, n, trim=0.1)
qrBenchmarkgputools(N, n, trim=0.1)
svdBenchmark(N, n, trim=0.1)
```

```
luBenchmark(N, n, trim=0.1)
luBenchmarkgputools(N, n, trim=0.1)
```

Arguments

N	dimension of square matrix
n	number of replications of benchmarked test
trim	percentage to be trimmed in <code>mean</code> estimation

Details

`getMatrix` provides a square matrix of the given dimension.

`matmultBenchmark` times the cost of multiplying a matrix of the given size with itself, repeated as specified and returns the trimmed mean of the elapsed times. `matmultBenchmarkgputools` does the same using the **gputools** and packages.

`qrBenchmark` times the cost of a QR decomposition of a matrix of the given size, repeated as specified and returns the trimmed mean of the elapsed times. `qrBenchmarkgputools` does the same using the **gputools** packages.

`svdBenchmark` times the cost of a Singular Value Decomposition (SVD) of a matrix of the given size, repeated as specified and returns the trimmed mean of the elapsed times.

`luBenchmark` times the cost of a LU Decomposition of a matrix of the given size, repeated as specified and returns the trimmed mean of the elapsed times. `luBenchmarkgputools` does the same using the **gputools** package.

figures

Figures from the corresponding vignette

Description

These functions generate the figures the in the corresponding vignette.

Usage

```
loglogAnalysis()  
figure_LU_i7(D)  
figure_LU_xeon(D)  
figure_MatMult_i7(D)  
figure_MatMult_xeon(D)  
figure_QR_i7(D)  
figure_QR_xeon(D)  
figure_SVD_i7(D)  
figure_SVD_xeon(D)  
figure_LogLogIntercept()  
figure_LogLogSlopes()  
figure_LogLogLattice(titles=TRUE)  
figure_Lattice(titles=TRUE)
```

Arguments

D	Benchmark results to be visualised
titles	Boolean flag whether to set ‘main’ and ‘sub’ titles for the figure

Details

`loglogAnalysis` retrieves past benchmark results from the database contained in the package and returns intercepts and slopes of regressions of elapsed times on matrix dimensions (where both inputs are in logarithms).

The various figure functions create the corresponding figures from the vignette.

 utilities

Utility functions for GPU/CPU Benchmarking

Description

Utility functions for GPU/CPU Benchmarking

Usage

```

requirements()

createDatabase(dbfile)
databaseResult(data,dbfile)

installAtlas()
installAtlas39()
installGoto()
installMKL()
purgeAtlas()
purgeAtlas39()
purgeGoto()
purgeMKL()

isPackageInstalled(package)
hasGputools()

getBenchmarkData(host)

```

Arguments

data	a (one-row) dataframe containing results from a benchmark
dbfile	character string containing path and name of SQLite database file
package	character string denoting a package to test for
host	character string denoting the host system for which benchmark data is to be retrieved

Details

`requirements` checks for a few system requirements such platform (Unix), operating system provider (Debian or Ubuntu) and presence of key packages (`gotoblas2-helper`).

`createDatabase` creates an empty SQLite database to store benchmark results.

`databaseResult` stores the benchmark results in the SQLite database.

The different `install*` functions add the respective BLAS libraries to the system; the different `purge*` functions do the inverse operation and remove them.

The function `hasGputools` tests for presence of this CRAN package on the current machine – as a very cheap proxy to testing whether the machine is GPU-capable or not. It uses the function `isPackageInstalled` for this test.

The function `getBenchmarkData` retrieves benchmark results for a given host.

Index

*Topic **misc**

- analysis, [2](#)
 - benchmark, [2](#)
 - figures, [3](#)
 - utilities, [4](#)
- analysis, [2](#)
- benchmark, [2](#)
- createDatabase (utilities), [4](#)
- databaseResult (utilities), [4](#)
- figure_Lattice (figures), [3](#)
- figure_LogLogIntercept (figures), [3](#)
- figure_LogLogLattice (figures), [3](#)
- figure_LogLogSlopes (figures), [3](#)
- figure_LU_i7 (figures), [3](#)
- figure_LU_xeon (figures), [3](#)
- figure_MatMult_i7 (figures), [3](#)
- figure_MatMult_xeon (figures), [3](#)
- figure_QR_i7 (figures), [3](#)
- figure_QR_xeon (figures), [3](#)
- figure_SVD_i7 (figures), [3](#)
- figure_SVD_xeon (figures), [3](#)
- figures, [3](#)
- getBenchmarkData (utilities), [4](#)
- getMatrix (benchmark), [2](#)
- hasGputools (utilities), [4](#)
- installAtlas (utilities), [4](#)
- installAtlas39 (utilities), [4](#)
- installGoto (utilities), [4](#)
- installMKL (utilities), [4](#)
- isPackageInstalled (utilities), [4](#)
- loglogAnalysis (analysis), [2](#)
- luBenchmark (benchmark), [2](#)
- luBenchmarkgputools (benchmark), [2](#)
- matmultBenchmark (benchmark), [2](#)
- matmultBenchmarkgputools (benchmark), [2](#)
- mean, [3](#)
- purgeAtlas (utilities), [4](#)
- purgeAtlas39 (utilities), [4](#)
- purgeGoto (utilities), [4](#)
- purgeMKL (utilities), [4](#)
- qrBenchmark (benchmark), [2](#)
- qrBenchmarkgputools (benchmark), [2](#)
- requirements (utilities), [4](#)
- svdBenchmark (benchmark), [2](#)
- utilities, [4](#)