Package 'gcbd'

October 30, 2024

Type Package

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

Version 0.2.7

Date 2024-10-23

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

URL https://github.com/eddelbuettel/gcbd

BugReports https://github.com/eddelbuettel/gcbd/issues

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

Author Dirk Eddelbuettel [aut, cre] (<https://orcid.org/0000-0001-6419-907X>)

Maintainer Dirk Eddelbuettel <edd@debian.org>

Repository CRAN

Date/Publication 2024-10-29 23:30:02 UTC

Contents

`analysis` *Analysis functions for GPU/CPU Benchmarking*

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 slops of regressions of elapsed times on matrix dimensions (where both inputs are in logarithms).

`benchmark` *Benchmarking functions for GPU/CPU Benchmarking*

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 mean 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

```
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

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 presense 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 in GPU-capable or not. It uses the function isPackageInstalled for this test.

The function getBenchmarkData retrieves benchmark results for a given host.

Index