Package ‘RcppParallel’

June 25, 2020

Type Package
Title Parallel Programming Tools for 'Rcpp'
Version 5.0.2
Description High level functions for parallel programming with 'Rcpp'.
For example, the 'parallelFor()' function can be used to convert the work of
a standard serial "for" loop into a parallel one and the 'parallelReduce()'
function can be used for accumulating aggregate or other values.
Depends R (>= 3.0.2)
Suggests Rcpp, RUnit, knitr, rmarkdown
SystemRequirements GNU make, Windows: cmd.exe and cscript.exe,
Solaris: g++ is required
License GPL-2
URL http://rcppcore.github.io/RcppParallel,
https://github.com/RcppCore/RcppParallel
BugReports https://github.com/RcppCore/RcppParallel/issues
Biarch TRUE
Collate 'build.R' 'hooks.R' 'options.R' 'skeleton.R'
NeedsCompilation yes
Author JJ Allaire [aut],
Romain Francois [aut, cph],
Kevin Ushey [aut, cre],
Gregory Vandenbrouck [aut],
Marcus Geelnard [aut, cph] (TinyThread library,
http://tinythreadpp.bitsnbites.eu/),
RStudio [cph],
Intel [aut, cph] (Intel TBB library,
https://www.threadingbuildingblocks.org/),
Microsoft [cph]
Maintainer Kevin Ushey <kevin@rstudio.com>
Repository CRAN
Date/Publication 2020-06-24 22:20:03 UTC
R topics documented:

- RcppParallel-package
- RcppParallel.package.skeleton
- RcppParallelFlags
- setThreadOptions

Description

High level functions for doing parallel programming with Rcpp. For example, the parallelFor function can be used to convert the work of a standard serial "for" loop into a parallel one and the parallelReduce function can be used for accumulating aggregate or other values.

The high level interface enables safe and robust parallel programming without direct manipulation of operating system threads. On Windows, OS X, and Linux systems the underlying implementation is based on Intel TBB (Threading Building Blocks). On other platforms a less-performant fallback implementation based on the TinyThread library is used.

For additional documentation see the package website at: http://rcppcore.github.io/RcppParallel.

Author(s)

JJ Allaire, Romain Francois, Gregory Vandenbrouck, Marcus Geelnard, Intel Inc.

Description

RcppParallel.package.skeleton automates the creation of a new source package that intends to use features of RcppParallel.

It is based on the package.skeleton function which it executes first.

Usage

RcppParallel.package.skeleton(
    name = "anRpackage",
    example_code = TRUE,
    ...
)
**Arguments**

- **name**
  The name of your R package.

- **example_code**
  If TRUE, example C++ code using RcppParallel is added to the package.

- **...**
  Optional arguments passed to Rcpp.package.skeleton.

**Details**

In addition to Rcpp.package.skeleton:

- The ‘DESCRIPTION’ file gains an Imports line requesting that the package depends on RcppParallel and a LinkingTo line so that the package finds RcppParallel header files.

- The ‘NAMESPACE’ gains a useDynLib directive as well as an importFrom(RcppParallel,evalCpp to ensure instantiation of RcppParallel.

- The ‘src’ directory is created if it does not exist and a ‘Makevars’ file is added setting the environment variables ‘PKG_LIBS’ to accommodate the necessary flags to link with the RcppParallel library.

- If the example_code argument is set to TRUE, example files ‘vector-sum.cpp’ is created in the ‘src’ directory. Rcpp::compileAttributes() is then called to generate src/RcppExports.cpp and R/RcppExports.R. These files are given as an example and should eventually be removed from the generated package.

**Value**

Nothing, used for its side effects

**References**

Read the Writing R Extensions manual for more details.

Once you have created a source package you need to install it: see the R Installation and Administration manual, INSTALL and install.packages.

**See Also**

package.skeleton

**Examples**

```r
## Not run:
# simple package
RcppParallel.package.skeleton( "foobar" )
```

## End(Not run)
**setThreadOptions**

**Description**

Set thread options (number of threads to use for task scheduling and stack size per-thread) for RcppParallel.

**Usage**

```r
setThreadOptions(numThreads = "auto",
                      stackSize = "auto")
```

**Arguments**

- `numThreads` : Number of threads to use for task scheduling (call defaultNumThreads to determine the default value used for "auto").
- `stackSize` : Stack size (in bytes) to use for worker threads. The default used for "auto" is 2MB on 32-bit systems and 4MB on 64-bit systems (note that this parameter has no effect on Windows).
setThreadOptions

Details

RcppParallel is automatically initialized with the default number of threads and thread stack size when it loads. You can call setThreadOptions at any time to change the defaults.

Value

The defaultNumThreads returns the default number of threads that are used by RcppParallel if another value isn’t specified using setThreadOptions.

Examples

```r
## Not run:
library(RcppParallel)
setThreadOptions(numThreads = 4)
defaultNumThreads()

## End(Not run)
```
Index

*Topic package
  RcppParallel-package, 2
*Topic parallel
  RcppParallel-package, 2
*Topic programming
  RcppParallel.package.skeleton, 2

CxxFlags (RcppParallelFlags), 4
defaultNumThreads (setThreadOptions), 4
INSTALL, 3
install.packages, 3

LdFlags (RcppParallelFlags), 4
package.skeleton, 2, 3

Rcpp.package.skeleton, 3
RcppParallel (RcppParallel-package), 2
RcppParallel-package, 2
RcppParallel.package.skeleton, 2
RcppParallelFlags, 4
RcppParallelLibs (RcppParallelFlags), 4
setThreadOptions, 4