Package ‘RcppParallel’

January 5, 2022

**Type** Package

**Title** Parallel Programming Tools for 'Rcpp'

**Version** 5.1.5

**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, Intel TBB, Windows: cmd.exe and cscript.exe, Solaris: g++ is required

**License** GPL (>= 3)

**URL** https://rcppcore.github.io/RcppParallel/, https://github.com/RcppCore/RcppParallel

**BugReports** https://github.com/RcppCore/RcppParallel/issues

**Biarch** TRUE

**RoxygenNote** 7.1.1

**Encoding** UTF-8

**NeedsCompilation** yes

**Author** JJ Allaire [aut],
               Romain Francois [aut, cph],
               Kevin Ushey [aut, cre],
               Gregory Vandenbrouck [aut],
               Marcus Geelnard [aut, cph] (TinyThread library, https://tinythreadpp.bitsnbites.eu/),
               Hamada S. Badr [ctb] (<https://orcid.org/0000-0002-9808-2344>),
               RStudio [cph],
               Intel [aut, cph] (Intel TBB library, https://www.threadingbuildingblocks.org/),
               Microsoft [cph]
**Maintainer**  Kevin Ushey <kevin@rstudio.com>

**Repository**  CRAN

**Date/Publication**  2022-01-05 10:50:07 UTC

---

**R topics documented:**

- `RcppParallel-package` ................................................................. 2
- `flags` ...................................................................................... 2
- `RcppParallel.package.skeleton` ...................................................... 3
- `setThreadOptions` ...................................................................... 4
- `tbbLibraryPath` .......................................................................... 5

**Index**  7

---

### `RcppParallel-package`

**Parallel programming tools for Rcpp**

**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.

**Details**

The high level interface enables safe and robust parallel programming without direct manipulation of operating system threads. On Windows, macOS, 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:

https://rcppcore.github.io/RcppParallel/

---

### `flags`

**Compilation flags for RcppParallel**

**Description**

Output the compiler or linker flags required to build against RcppParallel.

**Usage**

```
CxxFlags()
LdFlags()
RcppParallelLibs()
```
Details

These functions are typically called from Makevars as follows:

```bash
PKG_LIBS += $(shell "${R_HOME}/bin/Rscript" -e "RcppParallel::LdFlags()")
```

On Windows, the flags ensure that the package links with the built-in TBB library. On Linux and macOS, the output is empty, because TBB is loaded dynamically on load by RcppParallel.

R packages using RcppParallel should also add the following to their NAMESPACE file:

```r
importFrom(RcppParallel, RcppParallelLibs)
```

This is necessary to ensure that RcppParallel (and so, TBB) is loaded and available.

Value

Returns NULL, invisibly. These functions are called for their side effects (writing the associated flags to stdout).

Description

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

Usage

```r
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

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

In addition to Rcpppackage.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

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

## End(Not run)
tbbLibraryPath

**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).

**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.

The `parallelFor()` and `parallelReduce()` also accept `numThreads` as an argument, if you’d like to control the number of threads specifically to be made available for a particular parallel function call. Note that this value is advisory, and TBB may choose a smaller number of threads if the number of requested threads cannot be honored on the system.

**Value**

`defaultNumThreads()` returns the default number of threads used by RcppParallel, if another value isn’t specified either via `setThreadOptions()` or explicitly in calls to `parallelFor()` and `parallelReduce()`.

**Examples**

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

## End(Not run)
```

---

tbbLibraryPath  Get the Path to a TBB Library

**Description**

Retrieve the path to a TBB library. This can be useful for R packages using RcppParallel that wish to use, or re-use, the version of TBB that RcppParallel has been configured to use.

**Usage**

tbbLibraryPath(name = NULL)
Arguments

name  The name of the TBB library to be resolved. Normally, this is one of tbb, tbbmalloc, or tbbmalloc_proxy. When NULL, the library path containing the TBB libraries is returned instead.
Index

* package
  RcppParallel-package, 2
* parallel
  RcppParallel-package, 2
* programming
  RcppParallel.package.skeleton, 3

CxxFlags (flags), 2

defaultNumThreads (setThreadOptions), 4

flags, 2

INSTALL, 4
install.packages, 4

LdFlags (flags), 2

package.skeleton, 3, 4

Rcpp.package.skeleton, 3
RcppParallel (RcppParallel-package), 2
RcppParallel-package, 2
RcppParallel.package.skeleton, 3
RcppParallelLibs (flags), 2

setThreadOptions, 4

tbbLibraryPath, 5