Package 'doParallel'

July 22, 2025

Type Package

Title Foreach Parallel Adaptor for the 'parallel' Package

Version 1.0.17

Description Provides a parallel backend for the %dopar% function using the parallel package.

Depends R (>= 2.14.0), foreach (>= 1.2.0), iterators (>= 1.0.0), parallel, utils

Suggests caret, mlbench, rpart, RUnit

Enhances compiler

License GPL-2

URL https://github.com/RevolutionAnalytics/doparallel

BugReports https://github.com/RevolutionAnalytics/doparallel/issues

NeedsCompilation no

Author Folashade Daniel [cre],
Microsoft Corporation [aut, cph],
Steve Weston [aut],
Dan Tenenbaum [ctb]

Maintainer Folashade Daniel <fdaniel@microsoft.com>

Repository CRAN

Date/Publication 2022-02-07 12:50:02 UTC

Contents

doParallel-package 2
registerDoParallel 2

Index **4**

doParallel-package *The doParallel Package*

Description

The doParallel package provides a parallel backend for the foreach/%dopar% function using the `parallel` package of R 2.14.0 and later.

Details

Further information is available in the following help topics:

`registerDoParallel`	register doParallel to be used by foreach/%dopar%

To see a tutorial introduction to the doParallel package, use `vignette("gettingstartedParallel")`. To see a tutorial introduction to the foreach package, use `vignette("foreach")`.

To see a demo of doParallel computing the sinc function, use `demo(sincParallel)`.

Some examples (in addition to those in the help pages) are included in the "examples" directory of the doParallel package. To list the files in the examples directory, use `list.files(system.file("examples", package="doParallel"))`. To run the bootstrap example, use `source(system.file("examples", "bootParallel.R", package="doParallel"))`. This is a simple benchmark, executing both sequentally and in parallel. There are many more examples that come with the foreach package, which will work with the doParallel package if it is registered as the parallel backend.

For a complete list of functions with individual help pages, use `library(help="doParallel")`.

registerDoParallel *registerDoParallel*

Description

The `registerDoParallel` function is used to register the parallel backend with the `foreach` package.

Usage

```
registerDoParallel(cl, cores=NULL, ...)
stopImplicitCluster()
```

Arguments

cl	A cluster object as returned by makeCluster, or the number of nodes to be created in the cluster. If not specified, on Windows a three worker cluster is created and used.
cores	The number of cores to use for parallel execution. If not specified, the number of cores is set to the value of options("cores"), if specified, or to one-half the number of cores detected by the parallel package.
...	Package options. Currently, only the nocompile option is supported. If nocompile is set to TRUE, compiler support is disabled.

Details

The parallel package from R 2.14.0 and later provides functions for parallel execution of R code on machines with multiple cores or processors or multiple computers. It is essentially a blend of the snow and multicore packages. By default, the doParallel package uses snow-like functionality. The snow-like functionality should work fine on Unix-like systems, but the multicore-like functionality is limited to a single sequential worker on Windows systems. On workstations with multiple cores running Unix-like operating systems, the system fork call is used to spawn copies of the current process.

The doParallel backend supports both multicore and snow options passed through the foreach function. The supported multicore options are preschedule, set.seed, silent, and cores, which are analogous to the similarly named arguments to mclapply, and are passed using the .options.multicore argument to foreach. The supported snow options are preschedule, which like its multicore analog can be used to chunk the tasks so that each worker gets a prescheduled chunk of tasks, and attachExportEnv, which can be used to attach the export environment in certain cases where R's lexical scoping is unable to find a needed export. The snow options are passed to foreach using the .options.snow argument.

The function stopImplicitCluster can be used in vignettes and other places where it is important to explicitly close the implicitly created cluster.

Examples

```
cl <- makePSOCKcluster(2)
registerDoParallel(cl)
m <- matrix(rnorm(9), 3, 3)
foreach(i=1:nrow(m), .combine=rbind)
stopCluster(cl)
```

Index

∗ **package**
doParallel-package, 2
∗ **utilities**
registerDoParallel, 2

doParallel *(*doParallel-package*)*, 2
doParallel-package, 2

mclapply, *3*

registerDoParallel, 2

stopImplicitCluster
*(*registerDoParallel*)*, 2