Package 'mappp'

July 22, 2025

Title Map in Parallel with Progress

Version 1.0.0

Description Provides one function, which is a wrapper around purrr::map() with some extras on top, including parallel computation, progress bars, error handling, and result caching.

License MIT + file LICENSE

Encoding UTF-8

Imports memoise, progress, pbmcapply, parallel, parallelly, purrr, rlang

RoxygenNote 7.1.1

URL https://github.com/cole-brokamp/mappp

BugReports https://github.com/cole-brokamp/mappp/issues

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Cole Brokamp [aut, cre]

Maintainer Cole Brokamp <cole.brokamp@gmail.com>

Repository CRAN

Date/Publication 2022-01-25 09:22:42 UTC

Contents

mappp *map in parallel with progress*

Description

This function is a wrapper around purrr::map() (which applies a function to each element of a list or atomic vector) with some extras on top, including parallel computation, progress bar, error handling, and result caching.

Usage

```
mappp(
  .x,
  .f,
  parallel = FALSE,
  cache = FALSE,
  cache_name = "cache",
  error_capture = TRUE,
  error_quiet = TRUE,
  num_cores = NULL
)
```

Arguments

.x	list or vector of objects to apply over
.f	function to apply; allows for compact anonymous functions (see rlang::as_function() for details)
parallel	logical; use parallel processing?
cache	defaults to FALSE, which means no cache used. If TRUE, cache the results locally in a folder named according to cache_name using the memoise package
cache_name	a character string to use a custom cache folder name (e.g. "my_cache"); defaults to "cache"
error_capture	apply function to all elements and return those that error as NA ; this also messages user with name/index of offending element and resulting error message
error_quiet	quiet individual error messages when capturing error messages? or show them as they occur?
num_cores	the number of cores used for parallel processing. Can be specified as an integer, or it will guess the number of cores available with parallelly::availableCores(). won't have an effect if parallel is FALSE

Details

mappp is designed for long computations and as such it always uses a progress bar, and always returns a list. Long computations shouldn't worry about being type strict; instead, extract results in the right type from the results list.

A progress bar will be shown in the terminal using an interactive R session or in an .Rout file, if using R CMD BATCH and submitting R scripts for non-interactive completion. Although R Studio supports the progress bar for single process workers, it has a problem showing the progress bar if using parallel processing (see the discussion at http://stackoverflow.com/questions/27314011/mcfork-in-rstudio). In this specific case (R Studio + parallel processing), text updates will be printed to the file '.progress'. Use a shell and 'tail -f .progress' to see the updates.

Value

a list the same length as `.x`

Examples

```
X <- list("x" = 100, "y" = "a", "z" = 200)
slow_log <- function(.x) {
  Sys.sleep(0.5)
  log(.x)
}
# by default returns NA on error
mappp(X, slow_log)
# when not using error, entire calculation will fail
mappp(X, slow_log, error_capture = FALSE)
# showing error messages when they occur rather than afterwards can be useful
# but will cause problems with progress bar displays
mappp(X, slow_log, error_quiet = FALSE)
```

Index