Package 'future.BatchJobs'

September 29, 2019

Version 0.16.2

Depends R (>= 3.2.0), future (>= 1.14.0)

Imports BatchJobs (>= 1.8), R.utils

Suggests future.apply, listenv, markdown, R.rsp

VignetteBuilder R.rsp

Title A Future API for Parallel and Distributed Processing using BatchJobs

Description Implementation of the Future API on top of the 'BatchJobs' package. This allows you to process futures, as defined by the 'future' package, in parallel out of the box, not only on your local machine or ad-hoc cluster of machines, but also via high-performance compute ('HPC') job schedulers such as 'LSF', 'OpenLava', 'Slurm', 'SGE', and 'TORQUE' / 'PBS', e.g. 'y <- future.apply::future_lapply(files, FUN = process)'. NOTE: The 'BatchJobs' package is deprecated in favor of the 'batchtools' package. Because of this, it is recommended to use the 'future.batchtools' package instead of this package.

License LGPL (>= 2.1)

LazyLoad TRUE

URL https://github.com/HenrikBengtsson/future.BatchJobs

BugReports https://github.com/HenrikBengtsson/future.BatchJobs/issues

RoxygenNote 6.1.1

NeedsCompilation no

Author Henrik Bengtsson [aut, cre, cph]

Maintainer Henrik Bengtsson <henrikb@braju.com>

Repository CRAN

Date/Publication 2019-09-29 12:40:02 UTC

R topics documented:

`batchjobs_custom` *BatchJobs conf futures*

Description

A conf BatchJobs future sources one or more BatchJobs configuration files (R source scripts) to define the BatchJobs configuration environment, e.g. '`.BatchJobs.R`'.

Usage

```
batchjobs_custom(expr, envir = parent.frame(), substitute = TRUE,
  globals = TRUE, label = "BatchJobs", conf = NULL,
  pathname = NULL, workers = Inf, resources = list(),
  job.delay = FALSE, ...)
```

Arguments

`expr`	The R expression to be evaluated
`envir`	The environment in which global environment should be located.
`substitute`	Controls whether `expr` should be `substitute()`:d or not.
`globals`	(optional) a logical, a character vector, a named list, or a Globals object. If TRUE, globals are identified by code inspection based on `expr` and `tweak` searching from environment `envir`. If FALSE, no globals are used. If a character vector, then globals are identified by lookup based their names `globals` searching from environment `envir`. If a named list or a Globals object, the globals are used as is.
`label`	(optional) Label of the future (where applicable, becomes the job name for most job schedulers).
`conf`	A BatchJobs configuration environment.
`pathname`	(alternative) Pathname to one or more BatchJobs configuration files to be loaded in order. If NULL, then the **BatchJobs** package will search for such configuration files.
`workers`	(optional) Additional specification for the backend workers. If NULL, the default is used.
`resources`	A named list passed to the BatchJobs template (available as variable `resources`).
`job.delay`	(optional) Passed as is to `submitJobs()`.
`...`	Additional arguments passed to `BatchJobsFuture()`.

Details

If conf is NULL (default), then the BatchJobs configuration will be created from a set of BatchJobs configuration files (R script files) as given by argument pathname. If none are specified (default), then **BatchJobs** is designed to use (in order) all of following configuration files (if they exist):

- system("etc","BatchJobs_global_config.R",package="BatchJobs")
- ~/.BatchJobs.R (in user's home directory)
- .BatchJobs.R (in the current directory)

Value

An object of class BatchJobsFuture.

---

batchjobs_local *BatchJobs local and interactive futures*

Description

A BatchJobs local future is an synchronous uniprocess future that will be evaluated in a background R session. A BatchJobs interactive future is an synchronous uniprocess future that will be evaluated in the current R session (and variables will be assigned to the calling environment rather than to a local one). Both types of futures will block until the futures are resolved.

Usage

```
batchjobs_local(expr, envir = parent.frame(), substitute = TRUE,
  globals = TRUE, label = "BatchJobs", workers = 1L,
  job.delay = FALSE, ...)
```

Arguments

expr	The R expression to be evaluated
envir	The environment in which global environment should be located.
substitute	Controls whether expr should be substitute():d or not.
globals	(optional) a logical, a character vector, a named list, or a Globals object. If TRUE, globals are identified by code inspection based on expr and tweak searching from environment envir. If FALSE, no globals are used. If a character vector, then globals are identified by lookup based their names globals searching from environment envir. If a named list or a Globals object, the globals are used as is.
label	(optional) Label of the future (where applicable, becomes the job name for most job schedulers).
workers	(optional) Additional specification for the BatchJobs backend.
job.delay	(optional) Passed as is to submitJobs().
...	Additional arguments passed to BatchJobsFuture().

Details

BatchJobs local futures rely on the BatchJobs backend set up by `makeClusterFunctionsLocal()` and BatchJobs interactive futures on the one set up by `makeClusterFunctionsInteractive()`. These are supported by all operating systems.

An alternative to BatchJobs local futures is to use cluster futures of the **future** package with a single local background session, i.e. `plan(cluster,workers="localhost")`.

An alternative to BatchJobs interactive futures is to use transparent futures of the **future** package.

Value

An object of class `BatchJobsFuture`.

Examples

```
## Use local BatchJobs futures
plan(batchjobs_local)

## A global variable
a <- 1

## Create explicit future
f <- future({
  b <- 3
  c <- 2
  a * b * c
})
v <- value(f)
print(v)


## Create implicit future
v %<-% {
  b <- 3
  c <- 2
  a * b * c
}
print(v)
```

`batchjobs_template` *BatchJobs LSF, OpenLava, SGE, Slurm and Torque futures*

Description

LSF, OpenLava, SGE, Slurm and Torque BatchJobs futures are asynchronous multiprocess futures that will be evaluated on a compute cluster via a job scheduler.

Usage

```
batchjobs_lsf(expr, envir = parent.frame(), substitute = TRUE,
  globals = TRUE, label = "BatchJobs", pathname = NULL,
  resources = list(), workers = Inf, job.delay = FALSE, ...)

batchjobs_openlava(expr, envir = parent.frame(), substitute = TRUE,
  globals = TRUE, label = "BatchJobs", pathname = NULL,
  resources = list(), workers = Inf, job.delay = FALSE, ...)

batchjobs_sge(expr, envir = parent.frame(), substitute = TRUE,
  globals = TRUE, label = "BatchJobs", pathname = NULL,
  resources = list(), workers = Inf, job.delay = FALSE, ...)

batchjobs_slurm(expr, envir = parent.frame(), substitute = TRUE,
  globals = TRUE, label = "BatchJobs", pathname = NULL,
  resources = list(), workers = Inf, job.delay = FALSE, ...)

batchjobs_torque(expr, envir = parent.frame(), substitute = TRUE,
  globals = TRUE, label = "BatchJobs", pathname = NULL,
  resources = list(), workers = Inf, job.delay = FALSE, ...)
```

Arguments

`expr`	The R expression to be evaluated
`envir`	The environment in which global environment should be located.
`substitute`	Controls whether `expr` should be `substitute()`:d or not.
`globals`	(optional) a logical, a character vector, a named list, or a Globals object. If TRUE, globals are identified by code inspection based on `expr` and `tweak` searching from environment `envir`. If FALSE, no globals are used. If a character vector, then globals are identified by lookup based their names `globals` searching from environment `envir`. If a named list or a Globals object, the globals are used as is.
`label`	(optional) Label of the future (where applicable, becomes the job name for most job schedulers).
`pathname`	A BatchJobs template file (**brew** formatted).
`resources`	A named list passed to the BatchJobs template (available as variable `resources`).
`workers`	(optional) Additional specification for the BatchJobs backend.
`job.delay`	(optional) Passed as is to `submitJobs()`.
`...`	Additional arguments passed to `BatchJobsFuture()`.

Details

These type of BatchJobs futures rely on BatchJobs backends set up using the following **BatchJobs** functions:

- `makeClusterFunctionsLSF()` for Load Sharing Facility (LSF)

- makeClusterFunctionsOpenLava() for OpenLava
- makeClusterFunctionsSGE() for Sun/Oracle Grid Engine (SGE)
- makeClusterFunctionsSLURM() for Slurm
- makeClusterFunctionsTorque() for TORQUE / PBS

Value

An object of class BatchJobsFuture.

---

future.BatchJobs *future.BatchJobs: A Future for BatchJobs*

Description

The **future.BatchJobs** package implements the Future API on top of **BatchJobs** such that futures can be resolved on for instance high-performance compute (HPC) clusters via job schedulers. The Future API is defined by the **future** package.

Details

To use BatchJobs futures, load **future.BatchJobs**, and select the type of future you wish to use via plan().

Examples

```
plan(batchjobs_local)
demo("mandelbrot", package="future", ask=FALSE)


## Use local BatchJobs futures
plan(batchjobs_local)

## A global variable
a <- 1

v %<-% {
  b <- 3
  c <- 2
  a * b * c
}

print(v)
```

Index