Package 'condor'

August 28, 2023

Version 2.1.0

Date 2023-08-28

Title Interact with 'Condor' from R via SSH

Depends ssh

Imports stats, utils

SystemRequirements htcondor

Description Interact with 'Condor' from R via SSH connection. Files are first uploaded from user machine to submitter machine, and the job is then submitted from the submitter machine to 'Condor'. Functions are provided to submit, list, and download 'Condor' jobs from R. 'Condor' is an open source high-throughput computing software framework for distributed parallelization of computationally intensive tasks.

License GPL-3

URL https://github.com/PacificCommunity/ofp-sam-condor, https://htcondor.org

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Arni Magnusson [aut, cre], Nan Yao [aut], Jemery Day [ctb], Thomas Teears [ctb]

Maintainer Arni Magnusson <thisisarni@gmail.com>

Repository CRAN

Date/Publication 2023-08-28 04:00:02 UTC

R topics documented:

---

condor-package *Interact with Condor from R via SSH*

Description

Interact with Condor from R via SSH connection. Files are first uploaded from user machine to submitter machine, and the job is then submitted from the submitter machine to Condor. Functions are provided to submit, list, and download Condor jobs from R.

Condor is an open source high-throughput computing software framework for distributed parallelization of computationally intensive tasks.

Details

Main interface:

condor_submit	submit
condor_q	list queue
condor_dir	list directories
condor_download	download

Stop and remove:

condor_rm	stop jobs
condor_rmdir	remove directories

Utilities:

condor_log	show log file
summary.condor_log	show log file summary
ssh_exec_stdout	execute command

Author(s)

Arni Magnusson and Nan Yao, with contributions by Jemery Day and Thomas Teears.

References

https://github.com/PacificCommunity/ofp-sam-condor

https://htcondor.org

See Also

condor uses the **ssh** package to connect to the Condor submitter machine.

`condor_dir` *Condor Directories*

Description

List Condor run directories, either on submitter machine or on a local drive.

Usage

```
condor_dir(top.dir = "condor", local.dir = NULL, pattern = "*",
  report = TRUE, sort = "job.id", session = NULL, ...)
```

Arguments

`top.dir`	top directory on submitter machine that contains Condor run directories.
`local.dir`	local directory to examine instead of `top.dir`.
`pattern`	regular expression identifying which run directories to show. The default is to show all directories inside `top.dir` or `local.dir`.
`report`	whether to return a detailed report of the run status in each directory.
`sort`	column name or column number used to sort the report data frame.
`session`	optional object of class `ssh_connect`.
`...`	passed to `grep`.

Details

If the user passes `top.dir` that resembles a Windows local directory (drive letter, colon, forward slash), it is automatically interpreted as a `local.dir`. In other words, `condor_dir("c:/myruns")` and `condor_dir(local.dir="c:/myruns")` are equivalent.

The default value of `session = NULL` looks for a `session` object in the user workspace. This allows the user to run Condor functions without explicitly specifying the `session`.

Value

A data frame containing details about each directory, or if `report = FALSE` a `character` vector of directory names.

Note

If there are many Condor run directories, the report generation can take substantial time (one SSH execution per run directory). To quickly return a vector of directory names, pass `report = FALSE`.

Author(s)

Arni Magnusson.

See Also

`condor_submit`, `condor_q`, `condor_dir`, and `condor_download` provide the main Condor interface.

`condor_rm` stops Condor jobs and `condor_rmdir` removes directories on the submitter machine.

`condor_log` and `summary.condor_log` are called to produce the detailed report if `report = TRUE`.

`condor-package` gives an overview of the package.

Examples

```
## Not run:

# General workflow
session <- ssh_connect("servername")

condor_submit()
condor_q()
condor_dir()
condor_download()  # after job has finished

# Alternatively, examine runs on local drive
condor_dir(local.dir="myruns")
condor_dir("c:/myruns")

## End(Not run)
```

---

condor_download *Condor Download*

Description

Download results from a Condor job.

Usage

```
condor_download(run.dir = NULL, local.dir = ".", top.dir = "condor",
  create.dir = FALSE, pattern = "End.tar.gz|condor.*(err|log|out)$",
  overwrite = FALSE, untar.end = TRUE, session = NULL)
```

Arguments

run.dir	name of a Condor run directory inside top.dir.
local.dir	local directory to download to.
top.dir	top directory on submitter machine that contains Condor run directories.
create.dir	whether to create local.dir if it does not exist.
pattern	regular expression identifying which result files to download. Passing pattern="*" will download all files.
overwrite	whether to overwrite local files if they already exist.
untar.end	whether to extract End.tar.gz into *local.dir* after downloading. (Ignored if a file named 'End.tar.gz' was not downloaded.)
session	optional object of class ssh_connect.

Details

The default value of run.dir = NULL looks for Condor job results in *top.dir*/*local.dir*. For example, if local.dir = "c:/yft/run01" then the default run.dir becomes "condor/run01".

The default value of pattern="End.tar.gz|condor.*(err|log|out)$" downloads End.tar.gz and Condor log files. For many analyses, it can be convenient to pack all results into End.tar.gz to make it easy to find, download, and manage output files.

The default value of session = NULL looks for a session object in the user workspace. This allows the user to run Condor functions without explicitly specifying the session.

Value

No return value, called for side effects.

Author(s)

Arni Magnusson.

See Also

condor_submit, condor_q, condor_dir, and condor_download provide the main Condor interface.

condor_rm stops Condor jobs and condor_rmdir removes directories on the submitter machine.

condor-package gives an overview of the package.

Examples

```
## Not run:

# General workflow
session <- ssh_connect("servername")

condor_submit()
condor_q()
```

```
condor_dir()
condor_download()  # after job has finished

# Alternatively, download specific run to specific folder
condor_download("01_this_model", "c:/myruns/01_this_model")

## End(Not run)
```

---

condor_log *Condor Log*

Description

Show Condor log file from a run directory, either on submitter machine or on a local drive.

Usage

```
condor_log(run.dir = ".", top.dir = "condor", local.dir = NULL,
  session = NULL)
```

Arguments

run.dir	name of a Condor run directory inside top.dir.
top.dir	top directory on submitter machine that contains Condor run directories.
local.dir	local directory to examine instead of *top.dir/run.dir*.
session	optional object of class ssh_connect.

Details

The default value of session = NULL looks for a session object in the user workspace. This allows the user to run Condor functions without explicitly specifying the session.

Value

Log file contents as an object of class condor_log.

The condor_log class is simply a "character" vector with a print.condor_log method.

Author(s)

Arni Magnusson.

See Also

summary.condor_log shows Condor log file summary.

condor_dir lists Condor directories.

condor-package gives an overview of the package.

Examples

```
## Not run:

# Examine log files on submitter machine
session <- ssh_connect("servername")

condor_dir()
condor_log()
summary(condor_log())

# Alternatively, examine log file on local drive
condor_dir(local.dir="c:/myruns")
condor_log(local.dir="c:/myruns/01_this_model")
summary(condor_log(local.dir="c:/myruns/01_this_model"))

## End(Not run)
```

---

condor_q *Condor Queue*

Description

List the Condor job queue.

Usage

```
condor_q(all = FALSE, count = FALSE, global = FALSE, user = "",
  session = NULL)

condor_qq(all = TRUE, count = TRUE, global = TRUE, user = "",
  session = NULL)
```

Arguments

all	whether to list jobs from all users.
count	whether to only show the number of jobs.
global	whether to list jobs submitted from all submitter machines.
user	username to list jobs submitted by a given user.
session	optional object of class ssh_connect.

Details

The default value of session = NULL looks for a session object in the user workspace. This allows the user to run Condor functions without explicitly specifying the session.

Value

Screen output from the condor_q shell command, or a table if count = TRUE.

Note

The condor_q R function has the same defaults as the condor_q shell command, listing only jobs that were submitted by the current user from the current submitter machine.

The condor_qq alternative is the same function but with different default argument values, convenient for a *quick* overview of the *queue*.

Author(s)

Arni Magnusson.

See Also

condor_submit, condor_q, condor_dir, and condor_download provide the main Condor interface.

condor_rm stops Condor jobs and condor_rmdir removes directories on the submitter machine.

condor-package gives an overview of the package.

Examples

```
## Not run:

# General workflow
session <- ssh_connect("servername")

condor_submit()
condor_q()
condor_dir()
condor_download()  # after job has finished

# Alternatively, list number of jobs being run by each user
condor_q(all=TRUE, count=TRUE)

## End(Not run)
```

---

condor_rm *Condor Remove*

Description

Stop Condor jobs.

Usage

```
condor_rm(job.id = NULL, all = FALSE, top.dir = "condor",
  session = NULL)
```

Arguments

job.id	a vector of integers or directory names, indicating Condor jobs to stop.
all	whether to stop all Condor jobs owned by user.
top.dir	top directory on submitter machine that contains Condor run directories.
session	optional object of class ssh_connect.

Details

The top.dir argument only has an effect when job.id is a vector of directory names. For example, condor_rm("01_this") will stop the Condor job corresponding to directory condor/01_this on the submitter machine.

The default value of session = NULL looks for a session object in the user workspace. This allows the user to run Condor functions without explicitly specifying the session.

Value

No return value, called for side effects.

Author(s)

Nan Yao and Arni Magnusson.

See Also

condor_submit, condor_q, condor_dir, and condor_download provide the main Condor interface.

condor_rm stops Condor jobs and condor_rmdir removes directories on the submitter machine.

condor-package gives an overview of the package.

Examples

```
## Not run:

# General workflow
session <- ssh_connect("servername")

condor_submit()
condor_q()
condor_dir()
condor_download()  # after job has finished

# Stop one or multiple jobs
condor_rm(123456)                  # stop one job (integer)
```

```
condor_rm(c(123456, 123789))        # stop two jobs (integers)
condor_rm("01_this")                # stop one job (dirname)
condor_rm(c("01_this", "02_that"))  # stop two jobs (dirnames)
condor_rm(all=TRUE)                 # stop all jobs

## End(Not run)
```

`condor_rmdir` *Condor Remove Directory*

Description

Remove directories on the submitter machine.

Usage

```
condor_rmdir(run.dir, top.dir = "condor", quiet = FALSE, session = NULL)
```

Arguments

`run.dir`	name of a Condor run directory inside `top.dir`.
`top.dir`	top directory on submitter machine that contains Condor run directories.
`quiet`	whether to suppress messages.
`session`	optional object of class `ssh_connect`.

Details

The default value of `session = NULL` looks for a `session` object in the user workspace. This allows the user to run Condor functions without explicitly specifying the `session`.

Value

No return value, called for side effects.

Author(s)

Arni Magnusson.

See Also

`condor_submit`, `condor_q`, `condor_dir`, and `condor_download` provide the main Condor interface.

`condor_rm` stops Condor jobs and `condor_rmdir` removes directories on the submitter machine.

`condor-package` gives an overview of the package.

Examples

```
## Not run:

# General workflow
session <- ssh_connect("servername")

condor_submit()
condor_q()
condor_dir()
condor_download()  # after job has finished

# Remove one or more directories
condor_rmdir("01_this")                 # remove ~/condor/01_this (one run)
condor_rmdir(c("01_this", "02_that"))   # remove two model runs inside condor
condor_rmdir("test_runs", top.dir=".")  # remove ~/my_runs (many subdirs)

## End(Not run)
```

`condor_submit` *Condor Submit*

Description

Submit a Condor job.

Usage

```
condor_submit(local.dir = ".", run.dir = NULL, top.dir = "condor",
  exclude = "condor_mfcl|tar.gz|End", session = NULL)
```

Arguments

`local.dir`	local directory containing a Condor `*.sub` file and any other files necessary to run the job.
`run.dir`	name of a Condor run directory to create inside `top.dir`.
`top.dir`	top directory on submitter machine that contains Condor run directories.
`exclude`	pattern identifying files in `local.dir` that should not be submitted to Condor.
`session`	optional object of class `ssh_connect`.

Details

The default value of `run.dir = NULL` runs the Condor job in *top.dir/local.dir*. For example, if `local.dir = "c:/yft/run01"` then the default `run.dir` becomes `"condor/run01"`.

It can be practical to organize Condor runs inside the default `top.dir = "condor"` directory, to keep Condor runs separate from other directories inside the user home. To organize Condor runs directly in the home folder on the submitter machine, pass `top.dir = ""`.

The default value of `session = NULL` looks for a `session` object in the user workspace. This allows the user to run Condor functions without explicitly specifying the `session`.

Value

Remote directory name with the job id as a name attribute.

Note

This function performs two core tasks: (1) upload files from `local.dir` to submitter machine, and (2) execute shell command `condor_submit` on submitter machine to launch the Condor job.

Author(s)

Arni Magnusson.

See Also

`condor_submit`, `condor_q`, `condor_dir`, and `condor_download` provide the main Condor interface.

`condor_rm` stops Condor jobs and `condor_rmdir` removes directories on the submitter machine.

`condor-package` gives an overview of the package.

Examples

```
## Not run:

# General workflow
session <- ssh_connect("servername")

condor_submit()
condor_q()
condor_dir()
condor_download()  # after job has finished

# Alternatively, submit a specific run
condor_submit("c:/myruns/01_this_model")

## End(Not run)
```

---

ssh_exec_stdout *Execute and Capture Standard Output*

Description

Call `ssh_exec_internal` and convert the standard output to characters.

Usage

```
ssh_exec_stdout(command, session = NULL, ...)
```

Arguments

command	command or script to execute.
session	optional object of class ssh_connect.
...	passed to ssh_exec_internal.

Details

The default value of session = NULL looks for a session object in the user workspace. This allows the user to run Condor functions without explicitly specifying the session.

Value

A "character" vector containing the standard output.

Author(s)

Arni Magnusson.

See Also

ssh_exec_wait runs a command or script and shows the standard output in the R console, while returning the exit status.

ssh_exec_internal runs a command or script and buffers the standard output into a raw vector.

condor-package gives an overview of the package.

Examples

```
## Not run:
session <- ssh_connect("servername")

ssh_exec_wait(session, "ls")             # returns 0
ssh_exec_internal(session, "ls")$stdout  # returns a raw vector
ssh_exec_stdout("ls")                    # returns directory names

## End(Not run)
```

---

summary.condor_log *Summary Condor Log*

Description

Produce a summary of a Condor log file.

Usage

```
## S3 method for class 'condor_log'
summary(object, ...)
```

Arguments

object	an object of class condor_log.
...	passed to round.

Value

Data frame with the following columns:

job.id	job id.
status	text indicating whether job status is submitted, executing, aborted, or finished.
submit.time	date and time when job was submitted.
runtime	total duration of a job.
disk	disk space used by job (MB).
memory	memory used by job (MB).

Author(s)

Arni Magnusson.

See Also

condor_log shows Condor log file.

condor-package gives an overview of the package.

Examples

```
## Not run:

# Examine log files on submitter machine
session <- ssh_connect("servername")

condor_dir()
condor_log()
summary(condor_log())

#' # Alternatively, examine log files on local drive
condor_dir(local.dir="c:/myruns")
condor_log(local.dir="c:/myruns/01_this_model")
summary(condor_log(local.dir="c:/myruns/01_this_model"))

## End(Not run)
```

Index