Package: future.p2p (via r-universe)

November 22, 2026

Title Parallel Future Backend for Peer-to-Peer Clusters using 'pico.sh'

Version 0.6.0-9007

Description Implementation of the Future API <doi:10.32614/RJ-2021-048> that resolves futures on a peer-to-peer ('P2P') cluster, where you and your friends share your spare compute resources with each other. The backend is fully compliant with the Future API, as validated by the 'future.tests' conformance test suite, meaning that code using futures gives identical results and behaves the same regardless of which future backend is used. A common use is to parallelize existing map-reduce calls via the 'futurize' package <doi:10.48550/arXiv.2601.17578>, e.g. 'plan(future.p2p::cluster)' followed by 'lapply(xs, fcn) |> futurize()' or 'purrr::map(xs, fcn) |> futurize()'.

Imports future (>= 1.80.0), processx, callr, utils

Suggests commonmark, base64enc

VignetteBuilder future.p2p

License Apache License (>= 2)

URL https://future.p2p.futureverse.org/

BugReports https://github.com/futureverse/future.p2p/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.1

Repository https://futureverse-devel.r-universe.dev

Date/Publication 2026-11-21 22:10:32 UTC

RemoteUrl https://github.com/futureverse/future.p2p

RemoteRef develop

RemoteSha d602f020fc6eff9f4a4ee136d1cd578eae112b9f

Contents

cluster 2
host_cluster 3
p2p_cluster_name 4
pico_username 4
worker 5
zzz-future.p2p.options 6

Index **7**

`cluster` *Peer-to-Peer (P2P) futures*

Description

WARNING: This function must never be called. It may only be used with `future::plan()`

Usage

```
cluster(
  cluster = p2p_cluster_name(),
  host = "pipe.pico.sh",
  ssh_args = NULL,
  ...
)
```

Arguments

`cluster`	The P2P cluster to connect to.
`host`	(character string) The hostname serving the pico service.
`ssh_args`	(character vector) Optional SSH arguments.
`...`	Not used.

Details

A P2P future is an asynchronous multiprocess future that will be evaluated in a background R session.

The Pico P2P future backend relies on Pico (1) to distribute futures among a peer-to-peer (P2P) cluster of R workers. Users with a Pico account can join the P2P cluster by being invited to a shared folder.

Users who wish to contribute their compute power to the P2P cluster should call `worker()`.

Users who wish to take advantage of the compute power of the P2P cluster should use `plan(future.p2p::cluster)`.

Value

An object of class `PicoP2PFuture`.

References

1. pico.sh, The ultimate ssh powered services for developers, https://pico.sh/.

Examples

```
# Connect to personal P2P cluster, which is automatically launched
plan(future.p2p::cluster)

## Create future
a <- 42
f <- future({ 2 * a })

## Get results
v <- value(f)
print(v)
```

`host_cluster` *Hosts a shared P2P cluster*

Description

Hosts a shared P2P cluster

Usage

```
host_cluster(
  cluster = p2p_cluster_name(users, host = host, ssh_args = ssh_args),
  users = character(0L),
  host = "pipe.pico.sh",
  ssh_args = NULL,
  duration = 14 * 24 * 60 * 60
)
```

Arguments

`cluster`	(character string) The name of the P2P cluster.
`users`	(character vector) Names of Pico users who should have access, in addition to the owner. The default is a personal cluster that only you have access to.
`host`	(character string) The hostname serving the pico service.
`ssh_args`	(character vector) Optional SSH arguments.
`duration`	Duration (in seconds) to offer this cluster.

Value

Returns nothing.

Examples

```
# Connect to personal P2P cluster, which is automatically launched
host_cluster(users = c("bob", "carol"))
```

`p2p_cluster_name` *Gets the default name of the P2P cluster*

Description

Gets the default name of the P2P cluster

Usage

```
p2p_cluster_name(users = character(0), ...)
```

Arguments

`users`	Users to have access to the cluster. This controls whether the default cluster names should be "personal" or "friends".
`...`	Passed as-is to `pico_username()`.

Value

`p2p_cluster_name()` returns R option `future.p2p.cluster`, if set. If not set, it returns `{pico_name}/personal` if `length(users) == 0`, otherwise `{pico_name}/friends`.

`pico_username` *Gets your pico.sh username*

Description

Gets your pico.sh username

Usage

```
pico_username(host = "pico.sh", ssh_args = NULL, timeout = 10)
```

Arguments

`host`	(character string) The hostname serving the pico service.
`ssh_args`	(character vector) Optional SSH arguments.
`timeout`	(numeric scalar) Maximum number of seconds before giving up.

Value

A character string or a timeout error.

worker *Launches a P2P worker and adds it to a P2P cluster*

Description

Launches a P2P worker and adds it to a P2P cluster

Usage

```
worker(
  cluster = p2p_cluster_name(host = host, ssh_args = ssh_args),
  host = "pipe.pico.sh",
  ssh_args = NULL,
  duration = 60 * 60,
  sandbox = FALSE
)
```

Arguments

cluster	The P2P cluster to contribute to.
host	(character string) The hostname serving the pico service.
ssh_args	(character vector) Optional SSH arguments.
duration	Duration (in seconds) to offer working on futures.
sandbox	If TRUE, the future is resolved in R WebAssembly (webR), using the external rw tool (1).

Value

Nothing.

Sequential, single-core processing by default

A P2P worker runs sequentially and is configured with a single CPU core to prevent nested parallelization.

References

1. rw: CLI for webR with Sandboxing Features, *under development*, https://github.com/futureverse/rw.

Examples

```
## Start a P2P cluster worker
future.p2p::worker()
```

`zzz-future.p2p.options` *Options used by future.p2p*

Description

Below are the R options and environment variables that are used by the **future.p2p** package.

WARNING: Note that the names and the default values of these options may change in future versions of the package. Please use with care until further notice.

Packages must not change these options

Just like for other R options, as a package developer you must *not* change any of the below options. Only the end-user should set these. If you find yourself having to tweak one of the options, make sure to undo your changes immediately afterward.

Options

'`future.p2p.wormhole`': (character) Specifies the absolute path to the `wormhole` executable. If not specified, a default one will be installed.

Options for debugging

'`future.p2p.debug`': (logical) If `TRUE`, extensive debug messages are generated.

Environment variables that set R options

All of the above R '`future.p2p.*`' options can be set by corresponding environment variable `R_FUTURE_P2P_*` *when the* **future.p2p** *package is loaded*. This means that those environment variables must be set before the **future.p2p** package is loaded in order to have an effect. For example, if `R_FUTURE_P2P_DEBUG=true`, then option '`future.p2p.debug`' is set to `TRUE` (logical).

Examples

```
# See debug messages
options(future.p2p.debug = TRUE)
```

Index

cluster, 2

future.p2p.debug
        (zzz-future.p2p.options), 6
future.p2p.options
        (zzz-future.p2p.options), 6
future.p2p.wormhole
        (zzz-future.p2p.options), 6
future::plan(), *2*

host_cluster, 3

p2p_cluster_name, 4
pico_username, 4
pico_username(), *4*

R_FUTURE_P2P_DEBUG
        (zzz-future.p2p.options), 6
R_FUTURE_P2P_WORMHOLE
        (zzz-future.p2p.options), 6

worker, 5
worker(), *2*

zzz-future.p2p.options, 6