
parmap Documentation

Release 1.6.0

Sergio Oller

Aug 18, 2022

Contents

1	Indices and tables	3
	Python Module Index	5
	Index	7

`parmap.map` (*function, iterable, *args, **kwargs*)

This function is equivalent to:

```
>>> [function(x, args[0], args[1],...) for x in iterable]
```

Parameters

- **pm_parallel** (*bool*) – Force parallelization on/off
- **pm_chunksize** (*int*) – see `multiprocessing.pool.Pool`
- **pm_pool** (*multiprocessing.pool.Pool*) – Pass an existing pool
- **pm_processes** (*int*) – Number of processes to use in the pool. See `multiprocessing.pool.Pool`
- **pm_pbar** (*bool or dict*) – Show progress bar with optional information

`parmap.starmap` (*function, iterables, *args, **kwargs*)

Equivalent to:

```
>>> return ([function(x1,x2,x3,..., args[0], args[1],...) for
>>>             (x1,x2,x3...) in iterable])
```

Parameters

- **pm_parallel** (*bool*) – Force parallelization on/off
- **pm_chunksize** (*int*) – see `multiprocessing.pool.Pool`
- **pm_pool** (*multiprocessing.pool.Pool*) – Pass an existing pool
- **pm_processes** (*int*) – Number of processes to use in the pool. See `multiprocessing.pool.Pool`
- **pm_pbar** (*bool or dict*) – Show progress bar with optional information

`parmap.map_async` (*function, iterable, *args, **kwargs*)

This function is the `multiprocessing.Pool.map_async` version that supports multiple arguments.

```
>>> [function(x, args[0], args[1],...) for x in iterable]
```

Parameters

- **pm_parallel** (*bool*) – Force parallelization on/off. If False, the function won't be asynchronous.
- **pm_chunksize** (*int*) – see `multiprocessing.pool.Pool`
- **pm_callback** (*function*) – see `multiprocessing.pool.Pool`
- **pm_error_callback** (*function*) – (not on python 2) see `multiprocessing.pool.Pool`
- **pm_pool** (*multiprocessing.pool.Pool*) – Pass an existing pool.
- **pm_processes** (*int*) – Number of processes to use in the pool. See `multiprocessing.pool.Pool`

`parmap.starmap_async` (*function, iterables, *args, **kwargs*)

This function is the `multiprocessing.Pool.starmap_async` version that supports multiple arguments.

```
>>> return ([function(x1,x2,x3,..., args[0], args[1],...) for
>>>             (x1,x2,x3...) in iterable])
```

Parameters

- **pm_parallel** (*bool*) – Force parallelization on/off. If False, the function won't be asynchronous.
- **pm_chunksize** (*int*) – see `multiprocessing.pool.Pool`
- **pm_callback** (*function*) – see `multiprocessing.pool.Pool`
- **pm_error_callback** (*function*) – see `multiprocessing.pool.Pool`
- **pm_pool** (*multiprocessing.pool.Pool*) – Pass an existing pool.
- **pm_processes** (*int*) – Number of processes to use in the pool. See `multiprocessing.pool.Pool`

CHAPTER 1

Indices and tables

- `genindex`
- `modindex`
- `search`

p

`parmap`, [1](#)

M

`map()` (*in module parmap*), 1

`map_async()` (*in module parmap*), 1

P

`parmap(module)`, 1

S

`starmap()` (*in module parmap*), 1

`starmap_async()` (*in module parmap*), 1