
Zwembad

Robin De Schepper

Mar 09, 2021

CONTENTS:

1	zwembad package	3
1.1	Submodules	3
1.2	zwembad.pool module	3
1.3	Module contents	4
2	Indices and tables	5
	Python Module Index	7
	Index	9

ZWEMBAD PACKAGE

1.1 Submodules

1.2 `zwembad.pool` module

```
class zwembad.pool.ExitObject
```

```
    Bases: object
```

Object returned from the context manager to all non-master processes. Any attribute access on this object will raise a `WorkerExitSuiteSignal` so that the context is exited.

```
    is_master()
```

```
    is_worker()
```

```
    workers_exit()
```

```
class zwembad.pool.MPIPoolExecutor (master=0, comm=None)
```

```
    Bases: concurrent.futures._base.Executor
```

MPI based Executor. Will use all available MPI processes to execute submissions to the pool. The MPI process with rank 0 will continue while all other ranks halt and

```
    property idling
```

```
    is_master()
```

```
    is_worker()
```

```
    map (fn, *iterables)
```

Submits jobs for as long as all `iterables` provide values and places the results in a list. The iterables are consumed greedily.

```
    shutdown()
```

Close the pool and tell all workers to stop their work loop

```
    property size
```

```
    submit (fn, *args, **kwargs)
```

Submit a task to the MPIPool. `fn(*args, **kwargs)` will be called on an MPI process meaning that all data must be communicable over the MPI communicator, which by default uses pickle.

Parameters `fn` (*callable*) – Function to call on the worker MPI process.

```
    workers_exit()
```

```
exception zwembad.pool.PoolGuardError
```

```
    Bases: Exception
```

This error is raised if a user forgets to guard their pool context with a `:method:`~pool.MPIPoolExecutor.workers_exit`` call.

exception `zwembad.pool.WorkerExitSuiteSignal`

Bases: `Exception`

This signal is raised when a worker needs to exit before executing the suite of a `with` statement that only the master should execute.

1.3 Module contents

Both the package and the docs are pretty minimalistic: You create an `pool.MPIPoolExecutor` and either `submit()` jobs to it or `map()` a series of jobs to a list.

INDICES AND TABLES

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

PYTHON MODULE INDEX

Z

zwembad, [4](#)
zwembad.pool, [3](#)

INDEX

E

`ExitObject` (class in `zwembad.pool`), 3

I

`idling()` (`zwembad.pool.MPIPoolExecutor` property), 3

`is_master()` (`zwembad.pool.ExitObject` method), 3

`is_master()` (`zwembad.pool.MPIPoolExecutor` method), 3

`is_worker()` (`zwembad.pool.ExitObject` method), 3

`is_worker()` (`zwembad.pool.MPIPoolExecutor` method), 3

M

`map()` (`zwembad.pool.MPIPoolExecutor` method), 3

module

`zwembad`, 4

`zwembad.pool`, 3

`MPIPoolExecutor` (class in `zwembad.pool`), 3

P

`PoolGuardError`, 3

S

`shutdown()` (`zwembad.pool.MPIPoolExecutor` method), 3

`size()` (`zwembad.pool.MPIPoolExecutor` property), 3

`submit()` (`zwembad.pool.MPIPoolExecutor` method), 3

W

`WorkerExitSuiteSignal`, 4

`workers_exit()` (`zwembad.pool.ExitObject` method), 3

`workers_exit()` (`zwembad.pool.MPIPoolExecutor` method), 3

Z

module

`zwembad`, 4

`zwembad.pool`

 module, 3