

# Python Multiprocessing as a Task Queue

January 07, 2014 by Gouthaman Balaraman (<http://karuth.com/author/gouthaman-balaraman.html>)

---

Use multiprocessing module as a task queue, and over come GIL in python.

When you have computationally intensive tasks in your website (or scripts), it is conventional to use a task queue such as Celery (<http://www.celeryproject.org/>). Using Celery requires some amount of setup and if you want to avoid, try using the following task queue based on the multiprocessing. Depending on the application at hand, Celery might be an overkill. An alternate approach is to use multiprocessing as a task queue.

Here is a simple introduction to multiprocessing:

```
from multiprocessing import Pool
def expensive_function(x):
    # do your expensive time consuming process
    return x*x
if __name__ == '__main__':
    # start 4 worker processes
    pool = Pool(processes=4)
    # evaluate "f(10)" asynchronously
    result = pool.apply_async(expensive_function, [10])
    print result.get(timeout=1)
```

The above snippet is copied from the multiprocessing documentation, and is fairly self explanatory. In the main block we start a pool of 4 processes. Then we asynchronously evaluate the `expensive_function`.

One can use the same idea for a website as shown below in the Flask app example:

```
from multiprocessing import Pool
from flask import Flask

app = Flask(__name__)
_pool = None

def expensive_function(x):
    # import packages that is used in this function
    # do your expensive time consuming process
    return x*x

@app.route('/expensive_calc/<int:x>')
def route_expcalc(x):
    f = _pool.apply_async(expensive_function,[x])
    r = f.get(timeout=2)
    return 'Result is %d'%r

if __name__=='__main__':
    _pool = Pool(processes=4)
    try:
        # insert production server deployment code
        app.run()
    except KeyboardInterrupt:
        _pool.close()
        _pool.join()
```

🔗 python (<http://karuth.com/tag/python.html>) programming (<http://karuth.com/tag/programming.html>)

0 Comments karuth

 Login ▾

Sort by Best ▾

Share  Favorite ★



Start the discussion...

Be the first to comment.

 Subscribe

 Add Disqus to your site

DISQUS

© Gouthaman Balaraman (<http://karuth.com>) Proudly powered by Pelican  
(<http://getpelican.com/>)