

Multiprocessing_tutorial

October 18, 2022

1 Multiprocessing in Python

1.0.1 Author: <https://github.com/Tsangares>

Common pitfalls of multiprocessing with python.

1.0.2 The Process and Queue

First we do the classic process and queue; the queue is

```
[1]: from multiprocessing import Queue, Process
```

Lets make the computaiton to do will be a random walk. We are going to use numpy to do this.

```
[2]: import numpy as np
import random,time
```

```
[3]: def random_walk(dimension=3,iterations=10_000):
    location = np.zeros(dimension)
    for i in range(iterations):
        location += np.array(random.choices([-1,1],k=dimension))
    print(f"We landed at {location}")
    return location
start = time.time()
random_walk(4,100)
print(f'It took {time.time() - start:.06f} seconds to generate this random_
↪walk.')
```

We landed at [-2. -30. -8. 16.]

It took 0.005899 seconds to generate this random walk.

1. Make a list to contain all the processes
2. Make the processes
3. Run the processes
4. Wait for them

```
[4]: processes = []
for _ in range(4):
    p = Process(target=random_walk)
    processes.append(p)
```

```

    p.start()
for p in processes:
    p.join()
print("Done.")

```

```

We landed at [ 26. -48. -10.]
We landed at [ 10. -18. -62.]
We landed at [ 80.  54. 120.]
We landed at [-50. -28.  56.]
Done.

```

1.0.3 Wait I want the results?

Thats why we need a queue

```

[5]: queue = Queue()

#Wrap the function to add the results in a queue
random_walk_with_queue = lambda d=3,k=10_000: queue.put(random_walk(d,k))

#Change the target!
processes = []
for _ in range(4):
    p = Process(target=random_walk_with_queue)
    processes.append(p)
    p.start()
for p in processes:
    p.join()
print("Done.")

```

```

We landed at [ 10.  84. -34.]
We landed at [118. -14.  -2.]
We landed at [-58.  72. 124.]
We landed at [ 86.  18. -34.]
Done.

```

Now lets unpack the queue print(queue)

```

[6]: print(queue)
walks = []
while not queue.empty():
    walks.append(queue.get())

print(walks)

```

```

<multiprocessing.queue.Queue object at 0x7f9cdfa38d90>
[array([ 10.,  84., -34.]), array([118., -14.,  -2.]), array([-58.,  72.,
124.]), array([ 86.,  18., -34.])]

```

1.1 Yo that all kinda sucks... is there a better way

1.1.1 Yes POOL

Now we do the same using a multiprocessing pool

```
[7]: from multiprocessing import Pool
```

With 4 processes get 100 random walks in 4 dimensions with 10,000 iterations. But now my function needs an iterator to give it an index (tell it when to stop).

```
[8]: def random_walk_indexed(index,dimension,iterations):  
      return random_walk(dimension,iterations)
```

```
[9]: #The second argument in starmap is the parameters  
from itertools import repeat  
  
dimension = 4  
iterations = 10_000  
vectors = 100  
  
with Pool(4) as pool:  
    vecs = pool.starmap(random_walk_indexed,  
        ↪zip(range(vectors),repeat(dimension),repeat(vectors)))
```

```
We landed at [ -6. -16.  4.  2.]  
We landed at [ 16.  8. -16. -10.]  
We landed at [ 2. -2. -8.  0.]  
We landed at [-6.  2. -4.  2.]  
We landed at [ -2.  0. -2. -12.]  
We landed at [ 10.  4. -20. -2.]  
We landed at [-14. -12. 10.  2.]  
We landed at [ 4. -2.  0.  8.]  
We landed at [-16. -6.  0. -16.]  
We landed at [10.  8. -8. -4.]  
We landed at [10. 20. 14.  2.]  
We landed at [ 6.  4. -4. 12.]  
We landed at [10.  8. 14. -2.]  
We landed at [-18. 16.  2. -8.]  
We landed at [-18. -8. 10.  8.]  
We landed at [-8. -2. -6. 16.]  
We landed at [12.  6. -6.  0.]  
We landed at [ -4. -2. -10. -2.]  
We landed at [18. -2. -6.  4.]  
We landed at [ 4. -2.  2. -8.]  
We landed at [ 4. -2. -6. -6.]  
We landed at [ 4. -22.  6. 12.]  
We landed at [-10. -12. -6. 16.]  
We landed at [20. -8. -6. 18.]
```

We landed at [-18. 4. 0. -2.]
 We landed at [-6. -12. 4. 2.]
 We landed at [12. 4. -8. 2.]
 We landed at [22. -4. -6. 2.]
 We landed at [-4. 10. 18. 2.]
 We landed at [-8. 4. 4. 8.]
 We landed at [8. 10. -12. -4.]
 We landed at [0. 8. -12. 6.]
 We landed at [14. 0. -20. -2.]
 We landed at [8. -14. 10. -6.]
 We landed at [6. 4. -14. 8.]
 We landed at [-4. -18. 2. -4.]
 We landed at [0. 2. 10. 10.]
 We landed at [-2. -2. -12. -4.]
 We landed at [10. 4. 4. 16.]
 We landed at [-2. -6. 4. -4.]
 We landed at [6. 12. 6. 2.]
 We landed at [-18. 2. 0. -4.]
 We landed at [14. 6. 6. -8.]
 We landed at [-10. -6. 2. 6.]
 We landed at [4. 12. 2. -10.]
 We landed at [-20. -12. 6. -4.]
 We landed at [2. 16. 0. 2.]
 We landed at [2. 6. -12. 4.]
 We landed at [8. -4. 6. 6.]
 We landed at [4. 6. -18. 4.]
 We landed at [-16. 4. -4. -4.]
 We landed at [0. 0. -2. -6.]
 We landed at [-2. -20. -6. 0.]
 We landed at [-2. 2. 12. 0.]
 We landed at [4. 4. 24. -6.]
 We landed at [-4. -14. -6. -2.]
 We landed at [2. -10. 0. 14.]
 We landed at [-16. -2. 6. -4.]
 We landed at [0. -12. -10. 8.]
 We landed at [0. 4. -14. -2.]
 We landed at [-24. 4. 12. -2.]
 We landed at [-2. -8. -2. 8.]
 We landed at [-6. 6. -12. -16.]
 We landed at [4. 12. -6. -2.]
 We landed at [8. 2. 4. 16.]
 We landed at [26. 14. 0. 10.]
 We landed at [2. -2. 16. 2.]
 We landed at [0. 6. 6. 4.]
 We landed at [2. 4. 8. -14.]
 We landed at [-8. -10. 8. -2.]
 We landed at [-10. 20. 14. 4.]
 We landed at [-16. -10. -2. 2.]

```

We landed at [-2.  6. 28.  0.]
We landed at [ 2. -2. 18. -8.]
We landed at [-12. 16.  0.  8.]
We landed at [ 4. 12. 10. 14.]
We landed at [  8. -14. -14.  2.]
We landed at [-4. 18. -10.  2.]
We landed at [  6. -4. -8. -12.]
We landed at [  2. 16. -16.  6.]
We landed at [-8. -16. -12.  2.]
We landed at [-14. -4. 16. -2.]
We landed at [ 6. 14.  0. -8.]
We landed at [  2. -4. 18. -24.]
We landed at [-10. -16. -14. -10.]
We landed at [-10. -12. -14. 14.]
We landed at [ 8.  6. 20. -8.]
We landed at [ 0.  4.  2. 14.]
We landed at [-4. -6.  6.  6.]
We landed at [-14.  2. 18. 18.]
We landed at [  2.  4.  0. -22.]
We landed at [-14.  4.  2. 12.]
We landed at [-4. -2.  4.  2.]
We landed at [-10.  4.  0.  6.]
We landed at [-2.  4.  4. 22.]
We landed at [-6. -2.  4.  2.]
We landed at [  8. -4. 12. -10.]
We landed at [-8.  8. -2.  4.]
We landed at [-22. -2. -6.  2.]
We landed at [  8. -10. 14. -4.]

```

Now we can just look at our results:

```
[10]: print(vects)
```

```

[array([-6., -16.,  4.,  2.]), array([-2.,  0., -2., -12.]), array([-16.,
-6.,  0., -16.]), array([10.,  8., 14., -2.]), array([12.,  6., -6.,  0.]),
array([ 4., -2.,  2., -8.]), array([-18.,  4.,  0., -2.]), array([-6.,  2.,
-4.,  2.]), array([ 4., -2.,  0.,  8.]), array([10., 20., 14.,  2.]),
array([-18., -8., 10.,  8.]), array([18., -2., -6.,  4.]), array([-10., -12.,
-6., 16.]), array([-6., -12.,  4.,  2.]), array([ 16.,  8., -16., -10.]),
array([-14., -12., 10.,  2.]), array([ 6.,  4., -4., 12.]), array([-8., -2.,
-6., 16.]), array([ 4., -22.,  6., 12.]), array([12.,  4., -8.,  2.]),
array([-8.,  4.,  4.,  8.]), array([ 2., -2., -8.,  0.]), array([ 10.,  4.,
-20., -2.]), array([10.,  8., -8., -4.]), array([-18., 16.,  2., -8.]),
array([-4., -2., -10., -2.]), array([ 4., -2., -6., -6.]), array([20., -8.,
-6., 18.]), array([22., -4., -6.,  2.]), array([ 0.,  8., -12.,  6.]),
array([ 6.,  4., -14.,  8.]), array([10.,  4.,  4., 16.]), array([14.,  6.,
6., -8.]), array([ 2., 16.,  0.,  2.]), array([-2., -20., -6.,  0.]),
array([-4., 10., 18.,  2.]), array([ 14.,  0., -20., -2.]), array([ 0.,  2.,

```

```

10., 10.]), array([ 6., 12., 6., 2.]), array([-10., -6., 2., 6.]),
array([ 2., 6., -12., 4.]), array([ 0., 0., -2., -6.]), array([ 8., 10.,
-12., -4.]), array([-4., -18., 2., -4.]), array([-18., 2., 0., -4.]),
array([-20., -12., 6., -4.]), array([ 4., 6., -18., 4.]), array([-2.,
2., 12., 0.]), array([-16., -2., 6., -4.]), array([ 8., -14., 10.,
-6.]), array([-2., -2., -12., -4.]), array([-2., -6., 4., -4.]), array([
4., 12., 2., -10.]), array([ 8., -4., 6., 6.]), array([-16., 4., -4.,
-4.]), array([ 4., 4., 24., -6.]), array([-4., -14., -6., -2.]), array([
0., 4., -14., -2.]), array([ 4., 12., -6., -2.]), array([ 2., -2., 16.,
2.]), array([-10., 20., 14., 4.]), array([-12., 16., 0., 8.]), array([
-4., 18., -10., 2.]), array([ 2., -10., 0., 14.]), array([-24., 4.,
12., -2.]), array([ 8., 2., 4., 16.]), array([ 2., 4., 8., -14.]),
array([ 2., -2., 18., -8.]), array([ 6., -4., -8., -12.]), array([ 2., -4.,
18., -24.]), array([ 0., -12., -10., 8.]), array([-2., -8., -2., 8.]),
array([26., 14., 0., 10.]), array([-8., -10., 8., -2.]), array([-2., 6.,
28., 0.]), array([ 8., -14., -14., 2.]), array([-8., -16., -12., 2.]),
array([-6., 6., -12., -16.]), array([0., 6., 6., 4.]), array([-16., -10.,
-2., 2.]), array([ 4., 12., 10., 14.]), array([ 2., 16., -16., 6.]),
array([ 6., 14., 0., -8.]), array([ 8., 6., 20., -8.]), array([-14., -4.,
16., -2.]), array([-10., -12., -14., 14.]), array([-14., 2., 18., 18.]),
array([-4., -2., 4., 2.]), array([-2., 4., 4., 22.]), array([ 8., -4.,
12., -10.]), array([-22., -2., -6., 2.]), array([-10., -16., -14., -10.]),
array([ 0., 4., 2., 14.]), array([ 2., 4., 0., -22.]), array([-10., 4.,
0., 6.]), array([-6., -2., 4., 2.]), array([-8., 8., -2., 4.]), array([
8., -10., 14., -4.]), array([-4., -6., 6., 6.]), array([-14., 4., 2.,
12.])]

```