

Code:

```
1 import numpy as np
2 from timeit import default_timer as timer
3
4 def VectorAdd(a,b,c):
5     for i in range(a.size):
6         c[i] = a[i] + b[i]
7
8 def main():
9     N = 32000000
10
11     A = np.ones(N, dtype=np.float32)
12     B = np.ones(N, dtype=np.float32)
13     C = np.zeros(N, dtype=np.float32)
14
15     start=timer()
16     VectorAdd(A, B, C)
17     vectoradd_time = timer() - start
18
19     print("C[:5] = " + str(C[:5]))
20     print("C[:5] = " + str(C[:5]))
21
22     print("VectorAdd took %f seconds" % vectoradd_time)
23
24 main()
```

Results:

```
C[:5] = [ 2.  2.  2.  2.  2.]
C[:5] = [ 2.  2.  2. ..., 2.  2.  2.]
VectorAdd took 7.991139 seconds
```

New Code:

```
1 import numpy as np
2 from timeit import default_timer as timer
3 from numba import vectorize
4
5 @vectorize(["float32(float32, float32)"], target='cuda')
6 def VectorAdd(a,b):
7     return a + b
8
9 def main():
10     N = 32000000
11
12     A = np.ones(N, dtype=np.float32)
13     B = np.ones(N, dtype=np.float32)
14     C = np.zeros(N, dtype=np.float32)
15
16     start=timer()
17     C = VectorAdd(A, B)
18     vectoradd_time = timer() - start
19
20     print("C[:5] = " + str(C[:5]))
21     print("C[:5] = " + str(C[:5]))
22
23     print("VectorAdd took %f seconds" % vectoradd_time)
24
25 main()
```

Results:

```
C[:5] = [ 2.  2.  2.  2.  2.]
C[:5] = [ 2.  2.  2. ...,  2.  2.  2.]
VectorAdd took 0.612048 seconds
```

GPU - .612(!)

Vs

CPU - 7.99