nouvelle multiplication strainchamps

Preprint · March 2021

CITATIONS
0

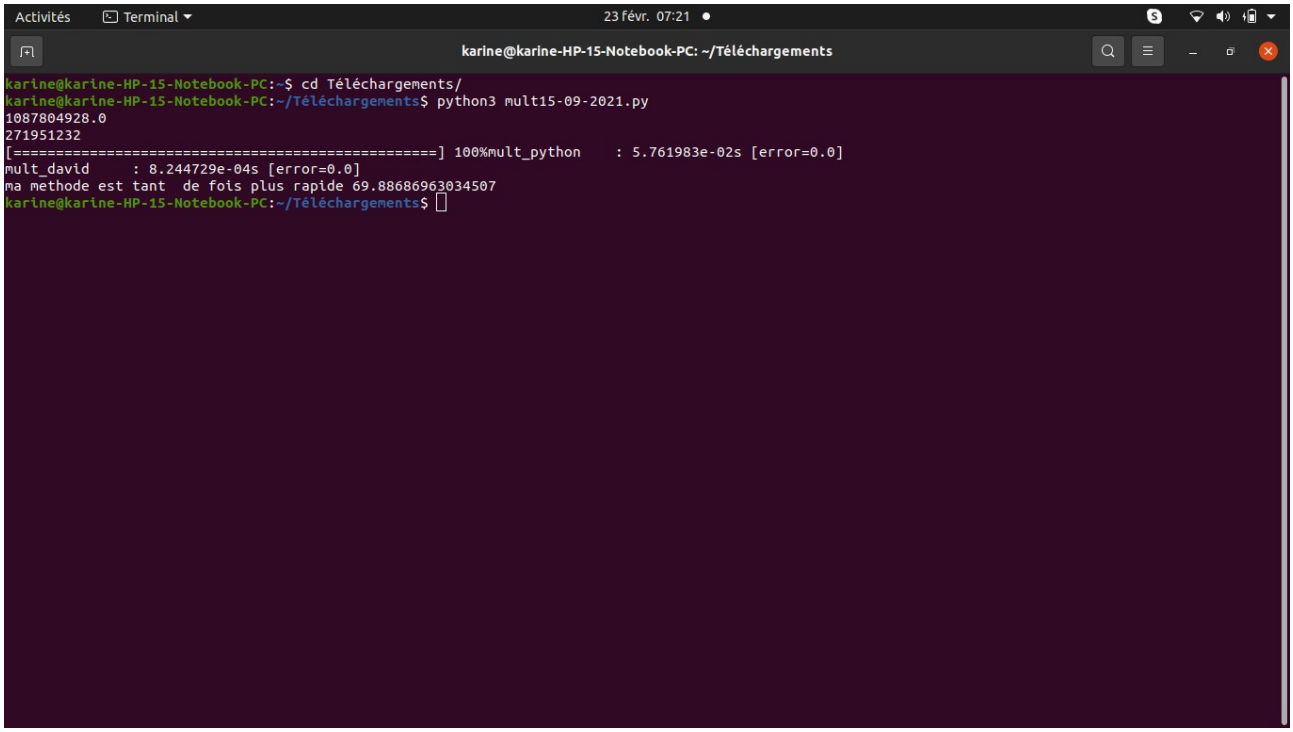
READS
113

1 author:

David Strainchamps
University of Burgundy
20 PUBLICATIONS **3** CITATIONS

SEE PROFILE

Le script python donne la résultat suivant que je n’arrive pas à reproduire sous C
Étonnant !!!




```
import gmpy2
from gmpy2 import mpz

import sys
import time

import numpy as np
from math import ceil, floor

n_tests = 200
g=4

bzero=1
czero=1
dzer=1



indice=np.array([1, 8, 11, 14, 20, 23, 26, 29, 35, 38, 41, 48, 50, 53, 60, 63, 68, 71, 74, 77, 83, 86, 89, 96, 98, 101, 108, 111, 113, 120, 123, 126, 131, 134, 137, 144, 146, 149, 156, 159, 161, 168, 171, 174, 180, 183, 186, 189, 194, 197, 204, 207, 209, 216, 219, 222, 228, 231, 234, 237, 243, 246, 249, 256])

v0=np.ones(64)
v1=indice
v2=indice**2
v3=indice**3
t2=mpz('0')
for i in range(64):
                        h=v1[i]
                        h2=h**2
                        h3=h**3
                        h=mpz(str(h))
                        h2=mpz(str(h2))
                        h3=mpz(str(h3))
                        t2=gmpy2.add(gmpy2.add(gmpy2.add(t2,h),h2),h3)
#                       t2=gmpy2.add(t2,h3)

a=256
b=256*257/2
c=np.sum((np.arange(1, (g ** g+1), 1))**2)
d=np.sum((np.arange(1, (g ** g+1), 1))**3)
t=b+c+d
X1=mpz('68647976601306097149819007990813932172694353001433054093944634591855431833976560521225596406614545549772963113914808580371219879997166438125740282911150571 51')**1457
X2=mpz('53113799281676709868958820655246862732959311772703192319944413820040355986085224273916250226522928566888932948624650101534657933765270723940951997876658735194383127083539321903172812 7')**16711
print(t)
print(t2)

def mult_python(x1, x2):
    return gmpy2.mul(x1,x2)



def mult_david(x1, x2):
    azero=gmpy2.div(gmpy2.sub(gmpy2.mul(x1,256),t),256)
    aprime=gmpy2.mul(azero,64)
    x=gmpy2.add(aprime,t2)
    azeroprime=gmpy2.div(gmpy2.sub(gmpy2.mul(x2,256),t),256)
    aprime=gmpy2.mul(azeroprime,64)
    y=gmpy2.add(aprime,t2)
    return gmpy2.div(gmpy2.mul(x,y),4096)


mult_functions = [mult_python, mult_david]

mult_times = {mult:[] for mult in mult_functions}
mult_errors = {mult:[] for mult in mult_functions}

N_ref = mult_python(X1,X2)


for i in range(n_tests):
    for mult in mult_functions:
        start_time = time.time()
        N = mult(X1, X2)
        end_time = time.time()
        mult_times[mult] += [end_time - start_time]
        mult_errors[mult] += [gmpy2.sub(N,N_ref)]
    sys.stdout.write('\r')
    sys.stdout.write("[{:50s}] {:d}%".format('='*(int(50*(i+1)/n_tests)), int(100*(i+1)/n_tests)))
    sys.stdout.flush()


for mult in mult_functions:
```

```
    print("{:15s}: {:e}s {error={}}".format(mult.__name__,
                                            np.mean(mult_times[mult]),
                                            np.mean(mult_errors[mult])))
print("ma methode est tant  de fois plus rapide",np.mean(mult_times[mult_python])/np.mean(mult_times[mult_david]))
```