# **CS 591 A1 Parallel Computing and Programming**

## **Assignment 1 --- Matrix Multiplication**

Members of group:

|  |  |  |
| --- | --- | --- |
| Xueyan Xia | U 82450191 | xueyanx@bu.edu |
| Ziqi Tan | U 88387934 | ziqi1756@bu.edu |

### **Requirements**

1. Implement parallel matrix multiplication of square matrices of 22d random integers, using 2k processes (i.e.., both matrix size and number of processes should be powers of 2 — d is the log2 of the number or rows/columns).
2. Compute the theoretical speedup with 1, 2, 4, and 8 processes, respectively (k = 0, 1, 2, and 3).
3. Test your program with various matrix sizes and numbers of workers and measure the actual running times.
4. Try to find the best performance (even if it is slower than the sequential version).
5. Compute speedup and efficiency for the best combination of matrix size and number of workers. Repeat this exercise for both threads (at the user level) and processes (at the kernel level).
6. Code must have complete docstrings, explaining the behavior of functions and the respective meanings of function parameters and return values. You must also use type hints on all symbols that allow them.

**Details**

1. Set your random seed to 1 so as to have a reproducible sequence of pseudo-random integers.

2. Your principal function should take k as one of its parameters.

3. You will need to work out some iteration logic to "chunk" your matrix appropriately into 2, 4, or 8 equal segments. (Look for tips in class.)

4. You will need to use both the threading and multiprocessing Python libraries for this exercise. Make separate versions, one using each of these libraries.

5. Use shared memory. For the process-based version you will find the multiprocessing. Array class useful for this purpose.

6. Use any Python construct for creating and destroying the threads and processes. You will likely find the lower-level Thread and Process classes the most useful for this assignment.

7. Test your program out on a CS Linux machine. If you do not yet have an account on one, please be sure to acquire one by registering here.

### **Methodology**

Assume we have matrix A and B with size of N by N where N is powers of 2.

Matrix C will be the result of A x B.

#### **Multiprocessing**

### **Result**

### **Analysis and Conclusion**

### **Result in CS machine**

**[ziqi1756@csa1 Assignment-1-Matrix-Multiplication]$ python3 multiprocessing\_matrix\_multiplication.py**

matrix size: 1

brute forcing...

brute force running time: 1.430511474609375e-05

number of processors: 1

parallel matrix multiplication...

actual number of processors: 1

multiprocessing running time: 0.017744779586791992

validating the result...

True

number of processors: 2

parallel matrix multiplication...

actual number of processors: 1

multiprocessing running time: 0.004422187805175781

validating the result...

True

number of processors: 4

parallel matrix multiplication...

actual number of processors: 1

multiprocessing running time: 0.004401683807373047

validating the result...

True

number of processors: 8

parallel matrix multiplication...

actual number of processors: 1

multiprocessing running time: 0.00426030158996582

validating the result...

True

summary

matrix size: 1

run time [(1, 0.017744779586791992), (1, 0.004422187805175781), (1, 0.004401683807373047), (1, 0.00426030158996582)]

speed up: [1.0, 4.012669829631227, 4.031361715957101, 4.165146342828362]

efficiency: [1.0, 4.012669829631227, 4.031361715957101, 4.165146342828362]

matrix size: 2

brute forcing...

brute force running time: 1.811981201171875e-05

number of processors: 1

parallel matrix multiplication...

actual number of processors: 1

multiprocessing running time: 0.004459381103515625

validating the result...

True

number of processors: 2

parallel matrix multiplication...

actual number of processors: 2

multiprocessing running time: 0.006894588470458984

validating the result...

True

number of processors: 4

parallel matrix multiplication...

actual number of processors: 4

multiprocessing running time: 0.008828401565551758

validating the result...

True

number of processors: 8

parallel matrix multiplication...

actual number of processors: 4

multiprocessing running time: 0.007523298263549805

validating the result...

True

summary

matrix size: 2

run time [(1, 0.004459381103515625), (2, 0.006894588470458984), (4, 0.008828401565551758), (4, 0.007523298263549805)]

speed up: [1.0, 0.646794384120617, 0.5051176105214832, 0.5927428299794011]

efficiency: [1.0, 0.3233971920603085, 0.1262794026303708, 0.14818570749485027]

matrix size: 4

brute forcing...

brute force running time: 3.3855438232421875e-05

number of processors: 1

parallel matrix multiplication...

actual number of processors: 1

multiprocessing running time: 0.0064792633056640625

validating the result...

True

number of processors: 2

parallel matrix multiplication...

actual number of processors: 2

multiprocessing running time: 0.005528926849365234

validating the result...

True

number of processors: 4

parallel matrix multiplication...

actual number of processors: 4

multiprocessing running time: 0.008667230606079102

validating the result...

True

number of processors: 8

parallel matrix multiplication...

actual number of processors: 8

multiprocessing running time: 0.011768817901611328

validating the result...

True

summary

matrix size: 4

run time [(1, 0.0064792633056640625), (2, 0.005528926849365234), (4, 0.008667230606079102), (8, 0.011768817901611328)]

speed up: [1.0, 1.171884432945235, 0.7475586609083157, 0.5505449536080386]

efficiency: [1.0, 0.5859422164726175, 0.18688966522707892, 0.06881811920100482]

matrix size: 8

brute forcing...

brute force running time: 0.00013494491577148438

number of processors: 1

parallel matrix multiplication...

actual number of processors: 1

multiprocessing running time: 0.006957054138183594

validating the result...

True

number of processors: 2

parallel matrix multiplication...

actual number of processors: 2

multiprocessing running time: 0.006610870361328125

validating the result...

True

number of processors: 4

parallel matrix multiplication...

actual number of processors: 4

multiprocessing running time: 0.008512735366821289

validating the result...

True

number of processors: 8

parallel matrix multiplication...

actual number of processors: 8

multiprocessing running time: 0.011617660522460938

validating the result...

True

summary

matrix size: 8

run time [(1, 0.006957054138183594), (2, 0.006610870361328125), (4, 0.008512735366821289), (8, 0.011617660522460938)]

speed up: [1.0, 1.0523658395845354, 0.8172524856462681, 0.5988343457560334]

efficiency: [1.0, 0.5261829197922677, 0.20431312141156702, 0.07485429321950418]

matrix size: 16

brute forcing...

brute force running time: 0.0008380413055419922

number of processors: 1

parallel matrix multiplication...

actual number of processors: 1

multiprocessing running time: 0.021717548370361328

validating the result...

True

number of processors: 2

parallel matrix multiplication...

actual number of processors: 2

multiprocessing running time: 0.015778064727783203

validating the result...

True

number of processors: 4

parallel matrix multiplication...

actual number of processors: 4

multiprocessing running time: 0.015370607376098633

validating the result...

True

number of processors: 8

parallel matrix multiplication...

actual number of processors: 8

multiprocessing running time: 0.01767134666442871

validating the result...

True

summary

matrix size: 16

run time [(1, 0.021717548370361328), (2, 0.015778064727783203), (4, 0.015370607376098633), (8, 0.01767134666442871)]

speed up: [1.0, 1.3764393000695094, 1.4129271432781647, 1.228969629919454]

efficiency: [1.0, 0.6882196500347547, 0.3532317858195412, 0.15362120373993174]

matrix size: 32

brute forcing...

brute force running time: 0.005939483642578125

number of processors: 1

parallel matrix multiplication...

actual number of processors: 1

multiprocessing running time: 0.13787364959716797

validating the result...

True

number of processors: 2

parallel matrix multiplication...

actual number of processors: 2

multiprocessing running time: 0.08051490783691406

validating the result...

True

number of processors: 4

parallel matrix multiplication...

actual number of processors: 4

multiprocessing running time: 0.09130048751831055

validating the result...

True

number of processors: 8

parallel matrix multiplication...

actual number of processors: 8

multiprocessing running time: 0.20778179168701172

validating the result...

True

summary

matrix size: 32

run time [(1, 0.13787364959716797), (2, 0.08051490783691406), (4, 0.09130048751831055), (8, 0.20778179168701172)]

speed up: [1.0, 1.7123990239973468, 1.510108580411655, 0.6635502008032128]

efficiency: [1.0, 0.8561995119986734, 0.37752714510291374, 0.0829437751004016]

matrix size: 64

brute forcing...

brute force running time: 0.04588794708251953

number of processors: 1

parallel matrix multiplication...

actual number of processors: 1

multiprocessing running time: 1.0575041770935059

validating the result...

True

number of processors: 2

parallel matrix multiplication...

actual number of processors: 2

multiprocessing running time: 0.6256117820739746

validating the result...

True

number of processors: 4

parallel matrix multiplication...

actual number of processors: 4

multiprocessing running time: 0.641352653503418

validating the result...

True

number of processors: 8

parallel matrix multiplication...

actual number of processors: 8

multiprocessing running time: 2.2767128944396973

validating the result...

True

summary

matrix size: 64

run time [(1, 1.0575041770935059), (2, 0.6256117820739746), (4, 0.641352653503418), (8, 2.2767128944396973)]

speed up: [1.0, 1.690352079987622, 1.6488653649701788, 0.46448727886427654]

efficiency: [1.0, 0.845176039993811, 0.4122163412425447, 0.05806090985803457]

matrix size: 128

brute forcing...

brute force running time: 0.374253511428833

number of processors: 1

parallel matrix multiplication...

actual number of processors: 1

multiprocessing running time: 8.01627516746521

validating the result...

True

number of processors: 2

parallel matrix multiplication...

actual number of processors: 2

multiprocessing running time: 4.656193733215332

validating the result...

True

number of processors: 4

parallel matrix multiplication...

actual number of processors: 4

multiprocessing running time: 4.890866279602051

validating the result...

True

number of processors: 8

parallel matrix multiplication...

actual number of processors: 8

multiprocessing running time: 19.20838475227356

validating the result...

True

summary

matrix size: 128

run time [(1, 8.01627516746521), (2, 4.656193733215332), (4, 4.890866279602051), (8, 19.20838475227356)]

speed up: [1.0, 1.7216369478530216, 1.639029715635051, 0.41733208027897195]

efficiency: [1.0, 0.8608184739265108, 0.40975742890876277, 0.052166510034871494]

**[ziqi1756@csa1 Assignment-1-Matrix-Multiplication]$ python3 multithreading\_matrix\_multiplication.py**

matrix size: 1

brute forcing...

brute force running time: 1.5020370483398438e-05

number of processors: 1

parallel matrix multiplication...

actual number of processors: 1

multithreading running time: 0.0003075599670410156

validating the result...

True

number of processors: 2

parallel matrix multiplication...

actual number of processors: 1

multithreading running time: 0.0003829002380371094

validating the result...

True

number of processors: 4

parallel matrix multiplication...

actual number of processors: 1

multithreading running time: 0.00019741058349609375

validating the result...

True

number of processors: 8

parallel matrix multiplication...

actual number of processors: 1

multithreading running time: 0.00015497207641601562

validating the result...

True

summary

matrix size: 1

run time [(1, 0.0003075599670410156), (1, 0.0003829002380371094), (1, 0.00019741058349609375), (1, 0.00015497207641601562)]

speed up: [1.0, 0.8032378580323786, 1.5579710144927537, 1.9846153846153847]

efficiency: [1.0, 0.8032378580323786, 1.5579710144927537, 1.9846153846153847]

matrix size: 2

brute forcing...

brute force running time: 1.2636184692382812e-05

number of processors: 1

parallel matrix multiplication...

actual number of processors: 1

multithreading running time: 0.0001537799835205078

validating the result...

True

number of processors: 2

parallel matrix multiplication...

actual number of processors: 2

multithreading running time: 0.00027108192443847656

validating the result...

True

number of processors: 4

parallel matrix multiplication...

actual number of processors: 4

multithreading running time: 0.00048542022705078125

validating the result...

True

number of processors: 8

parallel matrix multiplication...

actual number of processors: 4

multithreading running time: 0.0005099773406982422

validating the result...

True

summary

matrix size: 2

run time [(1, 0.0001537799835205078), (2, 0.00027108192443847656), (4, 0.00048542022705078125), (4, 0.0005099773406982422)]

speed up: [1.0, 0.5672823218997362, 0.31679764243614933, 0.3015427769985975]

efficiency: [1.0, 0.2836411609498681, 0.07919941060903733, 0.07538569424964937]

matrix size: 4

brute forcing...

brute force running time: 2.7418136596679688e-05

number of processors: 1

parallel matrix multiplication...

actual number of processors: 1

multithreading running time: 0.00022554397583007812

validating the result...

True

number of processors: 2

parallel matrix multiplication...

actual number of processors: 2

multithreading running time: 0.0002799034118652344

validating the result...

True

number of processors: 4

parallel matrix multiplication...

actual number of processors: 4

multithreading running time: 0.0005431175231933594

validating the result...

True

number of processors: 8

parallel matrix multiplication...

actual number of processors: 8

multithreading running time: 0.0010445117950439453

validating the result...

True

summary

matrix size: 4

run time [(1, 0.00022554397583007812), (2, 0.0002799034118652344), (4, 0.0005431175231933594), (8, 0.0010445117950439453)]

speed up: [1.0, 0.8057921635434412, 0.41527655838454786, 0.21593243551700525]

efficiency: [1.0, 0.4028960817717206, 0.10381913959613696, 0.026991554439625656]

matrix size: 8

brute forcing...

brute force running time: 0.00012540817260742188

number of processors: 1

parallel matrix multiplication...

actual number of processors: 1

multithreading running time: 0.0003917217254638672

validating the result...

True

number of processors: 2

parallel matrix multiplication...

actual number of processors: 2

multithreading running time: 0.0005478858947753906

validating the result...

True

number of processors: 4

parallel matrix multiplication...

actual number of processors: 4

multithreading running time: 0.0007524490356445312

validating the result...

True

number of processors: 8

parallel matrix multiplication...

actual number of processors: 8

multithreading running time: 0.0011439323425292969

validating the result...

True

summary

matrix size: 8

run time [(1, 0.0003917217254638672), (2, 0.0005478858947753906), (4, 0.0007524490356445312), (8, 0.0011439323425292969)]

speed up: [1.0, 0.7149695387293299, 0.520595690747782, 0.34243434764485203]

efficiency: [1.0, 0.35748476936466494, 0.1301489226869455, 0.042804293455606504]

matrix size: 16

brute forcing...

brute force running time: 0.0008182525634765625

number of processors: 1

parallel matrix multiplication...

actual number of processors: 1

multithreading running time: 0.001971006393432617

validating the result...

True

number of processors: 2

parallel matrix multiplication...

actual number of processors: 2

multithreading running time: 0.002070903778076172

validating the result...

True

number of processors: 4

parallel matrix multiplication...

actual number of processors: 4

multithreading running time: 0.002244710922241211

validating the result...

True

number of processors: 8

parallel matrix multiplication...

actual number of processors: 8

multithreading running time: 0.002809286117553711

validating the result...

True

summary

matrix size: 16

run time [(1, 0.001971006393432617), (2, 0.002070903778076172), (4, 0.002244710922241211), (8, 0.002809286117553711)]

speed up: [1.0, 0.951761455215289, 0.8780669144981412, 0.7016040057710261]

efficiency: [1.0, 0.4758807276076445, 0.2195167286245353, 0.08770050072137826]

matrix size: 32

brute forcing...

brute force running time: 0.006018400192260742

number of processors: 1

parallel matrix multiplication...

actual number of processors: 1

multithreading running time: 0.015001296997070312

validating the result...

True

number of processors: 2

parallel matrix multiplication...

actual number of processors: 2

multithreading running time: 0.014823198318481445

validating the result...

True

number of processors: 4

parallel matrix multiplication...

actual number of processors: 4

multithreading running time: 0.013681411743164062

validating the result...

True

number of processors: 8

parallel matrix multiplication...

actual number of processors: 8

multithreading running time: 0.015772581100463867

validating the result...

True

summary

matrix size: 32

run time [(1, 0.015001296997070312), (2, 0.014823198318481445), (4, 0.013681411743164062), (8, 0.015772581100463867)]

speed up: [1.0, 1.012014861756711, 1.096472884427715, 0.9510996901216839]

efficiency: [1.0, 0.5060074308783555, 0.27411822110692874, 0.11888746126521049]

matrix size: 64

brute forcing...

brute force running time: 0.047434329986572266

number of processors: 1

parallel matrix multiplication...

actual number of processors: 1

multithreading running time: 0.13367486000061035

validating the result...

True

number of processors: 2

parallel matrix multiplication...

actual number of processors: 2

multithreading running time: 0.12688231468200684

validating the result...

True

number of processors: 4

parallel matrix multiplication...

actual number of processors: 4

multithreading running time: 0.1212308406829834

validating the result...

True

number of processors: 8

parallel matrix multiplication...

actual number of processors: 8

multithreading running time: 0.11884856224060059

validating the result...

True

summary

matrix size: 64

run time [(1, 0.13367486000061035), (2, 0.12688231468200684), (4, 0.1212308406829834), (8, 0.11884856224060059)]

speed up: [1.0, 1.0535342166134583, 1.1026473069684293, 1.124749491962679]

efficiency: [1.0, 0.5267671083067291, 0.2756618267421073, 0.14059368649533488]

matrix size: 128

brute forcing...

brute force running time: 0.36438798904418945

number of processors: 1

parallel matrix multiplication...

actual number of processors: 1

multithreading running time: 1.0173559188842773

validating the result...

True

number of processors: 2

parallel matrix multiplication...

actual number of processors: 2

multithreading running time: 1.0700581073760986

validating the result...

True

number of processors: 4

parallel matrix multiplication...

actual number of processors: 4

multithreading running time: 1.070342779159546

validating the result...

True

number of processors: 8

parallel matrix multiplication...

actual number of processors: 8

multithreading running time: 1.0676295757293701

validating the result...

True

summary

matrix size: 128

run time [(1, 1.0173559188842773), (2, 1.0700581073760986), (4, 1.070342779159546), (8, 1.0676295757293701)]

speed up: [1.0, 0.9507482928931281, 0.9504954288411467, 0.9529109552714036]

efficiency: [1.0, 0.47537414644656406, 0.23762385721028667, 0.11911386940892545]

matrix size: 256

brute forcing...

brute force running time: 3.193373441696167

number of processors: 1

parallel matrix multiplication...

actual number of processors: 1

multithreading running time: 8.49372148513794

validating the result...

True

number of processors: 2

parallel matrix multiplication...

actual number of processors: 2

multithreading running time: 9.12512731552124

validating the result...

True

number of processors: 4

parallel matrix multiplication...

actual number of processors: 4

multithreading running time: 9.262306928634644

validating the result...

True

number of processors: 8

parallel matrix multiplication...

actual number of processors: 8

multithreading running time: 8.990013122558594

validating the result...

True

summary

matrix size: 256

run time [(1, 8.49372148513794), (2, 9.12512731552124), (4, 9.262306928634644), (8, 8.990013122558594)]

speed up: [1.0, 0.9308058059300366, 0.9170200847997594, 0.9447952265858977]

efficiency: [1.0, 0.4654029029650183, 0.22925502119993985, 0.1180994033232372]

matrix size: 512

brute forcing...

brute force running time: 37.5497841835022

number of processors: 1

parallel matrix multiplication...

actual number of processors: 1

multithreading running time: 79.369961977005

validating the result...

True

number of processors: 2

parallel matrix multiplication...

actual number of processors: 2

multithreading running time: 80.47326278686523

validating the result...

True

number of processors: 4

parallel matrix multiplication...

actual number of processors: 4

multithreading running time: 83.42097020149231

validating the result...

True

number of processors: 8

parallel matrix multiplication...

actual number of processors: 8

multithreading running time: 83.41899728775024

validating the result...

True

summary

matrix size: 512

run time [(1, 79.369961977005), (2, 80.47326278686523), (4, 83.42097020149231), (8, 83.41899728775024)]

speed up: [1.0, 0.9862898461967133, 0.9514389701450051, 0.9514614722977517]

efficiency: [1.0, 0.49314492309835667, 0.23785974253625128, 0.11893268403721896]