Registration Information:

- **Registration Number:** RA2111003010596
- **Name:** Pulkit Shringi
- **Section:** I1
- **Email:** ph1973@srmist.edu.in

1. Data regions and Data clauses:

```
import numpy as np
from numba import cuda

# Define kernel function
@cuda.jit
def add_arrays(a, b, c):
    idx = cuda.grid(1)
    if idx < len(a):
        c[idx] = a[idx] + b[idx]

# Main function
def main():
    # Array size
    N = 1000

    # Initialize input arrays on the CPU
    a = np.random.rand(N).astype(np.float32)
    b = np.random.rand(N).astype(np.float32)

    # Allocate output array on the CPU
    c = np.zeros_like(a)

    # Allocate memory on the GPU and transfer data
    d_a = cuda.to_device(a)
    d_b = cuda.to_device(b)
    d_c = cuda.to_device(c)

    # Launch kernel with data regions and data clauses
    with cuda.gpus[0]:
        add_arrays[N, 1](d_a, d_b, d_c)

    # Transfer result back to the CPU
    d_c.copy_to_host(c)

    # Print result
    print("Result array (c):", c)

if __name__ == "__main__":
    main()
```

```
Result array (c): [0.6096229  0.8598883  0.5664964  0.34593064
1.0582803  1.0863649
 1.4230244  1.277323   0.9146615  1.4402969  1.2720772  0.5846211
 0.9630954  1.7520881  1.8162138  1.2655301  0.40332395 0.29248813
 0.9078927  1.0974655  1.2489278  1.5679469  1.0475826  0.9132792
 0.7656118  1.5614353  0.921546   1.4091947  1.6545918  1.1901544
 1.1047037  1.4551964  0.5218992  0.37507898 0.14032298 0.4797646
 0.8149321  1.758665   1.0074725  0.6575542  0.6529774  0.79849875
 1.2796831  0.9507731  0.6630018  1.1136363  1.4834213  1.0963898
 0.08343747 1.6607887  1.1754173  0.5026342  1.3812509  1.3086209
 1.1488967  1.650713   0.2926948  1.7733827  1.2763078  0.6931138
 0.30873072 1.2590013  0.9102328  0.7219488  1.0180261  0.7449136
 0.6055236  1.4622303  1.4683558  1.2637503  1.3511887  0.812295
 0.95701796 0.5314009  1.184686   0.9493573  0.9276598  0.7817453
 1.4221996  0.9534637  1.1939527  1.4777131  1.0108751  1.3441068
 0.5606459  1.491529   0.8740469  0.40571907 0.5913063  0.2239683
 1.1175126  1.0941362  0.8761736  0.86667097 1.2759584  0.9511149
 0.6853385  1.4043171  0.8154901  0.72991824 0.34559986 0.3374446
 0.7855311  1.0529758  0.50753486 0.34788895 1.6471186  1.1029897
 1.0086973  1.3609977  0.8417458  0.7833881  1.1772933  1.4403872
 0.9543474  0.5184753  0.83854145 0.19835246 1.1769875  0.92235595
 0.8064274  1.3853089  0.90774906 1.1815124  0.92123145 0.67156035
 1.2134933  1.1166475  0.95590496 1.3081751  0.79894114 1.2765695
 1.7014492  1.1263226  0.8849149  0.62054133 1.2517405  1.5440621
 0.87382984 0.5172915  0.47438118 1.0217137  1.3135871  1.4159236
 1.1067119  0.86436343 0.9120478  0.65884054 1.6363828  0.84039694
 0.5267421  1.3009096  1.6434159  1.0486614  1.3609426  1.6053509
 1.1971748  1.3532934  1.2557331  0.6316004  0.64499456 1.4286163
 0.5646994  1.4927835  0.36860263 0.59011054 0.44381636 0.99749523
 0.96326536 0.8853983  1.7762651  1.4603271  0.51183426 0.49335682
 0.53595555 1.413887   1.2228285  1.5057027  1.2472683  1.0880591
 1.729038   0.9034805  1.1406617  1.0392075  1.2766144  1.5143259
 1.0204628  1.3310657  1.0417666  1.1870233  1.5616474  1.0792177
 0.6645834  1.1932795  1.1264808  0.6292772  1.6196581  1.1641927
 1.1476772  1.6665614  0.12442356 0.6532001  0.73313963 1.1537414
 1.5186367  1.4214909  0.6709719  0.55423784 0.468963   1.0620263
 1.0790803  0.5998752  1.1371291  1.6897995  0.69303495 0.6292792
 0.85887074 1.0628831  0.6749141  1.4683075  0.85186905 0.88367665
 0.68524766 0.33425835 1.2908111  0.1866799  0.59986836 0.6558503
 1.4519889  0.7324306  1.7708735  0.6620484  1.7045376  1.3376119
 0.81105065 1.6681747  1.1305313  0.8646219  0.8839829  1.1226976
 1.7639527  1.0647727  1.2004306  0.971313   0.7132802  1.096679
 1.0930498  0.9339528  1.0165689  1.2440276  1.39346    1.3804691
 1.37905    1.120745   1.2974975  1.8899455  1.5632391  1.0810843
 0.64242625 1.5818784  0.90261495 1.270538   0.21493167 0.82406634
 1.1942081  1.1802866  0.9795438  1.7266023  0.6862961  0.9322737
 0.85657537 0.6864082  0.37821364 1.4212892  1.3931942  0.8942565
 1.0309525  1.0814602  0.49840838 0.42530483 0.89126897 0.58180404
 0.18284267 0.59079945 0.7222342  1.3315462  1.0309764  1.2144215
 1.4105518  1.1069727  1.1039023  1.133691   1.7999866  0.9708021
```

0.45053202 0.7137438 0.6147286 1.8500948 0.9393273 1.3741744
1.2905753 1.2661654 1.623445   0.3577953 0.5073223 0.89418787
1.5059571 0.79580414 1.6493183 0.942711   0.94169366 1.1878504
1.2353234 0.65325934 0.35407823 0.6151955 1.0491261 1.0940962
1.3706111 0.2494082 1.499784   0.4517435 0.5917506 1.4392684
0.5179852 0.68631816 0.28913355 0.8635075 0.26857293 1.4600818
1.6028352 0.9662287 0.89846486 0.38118958 0.93900615 0.8924652
0.965816   0.65426004 0.4028138 0.17160285 0.8494575 0.34994572
1.7718503 1.758952   1.1669309 0.49608767 0.6481106 1.3781829
1.8029859 0.45758614 0.87455195 0.9303312 1.5738325 0.43340826
1.830466   0.97883475 1.2549301 1.6515532 1.0773048 0.9416307
1.9107275 1.2957788 1.1122873 1.3845751 1.0807014 0.3948313
0.87425923 0.50385654 1.1757178 0.4888128 1.2927767 0.9833944
1.2967423 0.80330247 0.99749017 1.1647747 1.610437   1.1723218
0.9965319 0.6020433 1.204448   0.5574684 1.8677568 0.9880817
0.6913567 0.18626171 1.4631145 1.2376299 1.4872233 1.7804881
0.87232816 1.1458806 1.2070793 0.7909813 1.5061665 1.2115842
1.4162409 1.0072898 1.4441707 0.8080151 0.11814032 0.91313887
0.81294906 0.16624036 1.0442263 0.4569077 1.5316617 0.76278234
1.6250436 1.5016282 1.2625555 0.56993204 1.3199432 1.738705
1.4553059 1.092717   1.1069342 0.89540726 0.85733795 1.0648713
1.172765   0.10113185 0.5288593 1.4427071 1.20124   0.7208482
0.7218858 1.0235702 0.96331495 1.1550401 0.81400347 1.6425934
0.85452056 1.4611903 1.3834305 0.74693274 1.2138429 0.45568362
0.6087817 0.8641237 0.9049148 0.8297179 0.783297   1.214545
1.2243149 1.3865877 1.0123217 0.83074415 0.63896805 0.81374943
1.1910176 0.6449773 1.1076083 0.5809461 0.7905412 0.9676435
0.39196414 1.3752735 1.1216803 0.5487388 1.5874147 0.3602377
1.3124487 0.78072315 1.0281563 0.32165003 1.2041502 1.0752318
0.96951425 0.95224285 0.7232282 1.1927068 1.190171   0.7349863
1.3044913 0.7009578 1.2287271 1.4696294 0.7904552 0.77339906
0.6755787 0.64614445 1.0545081 0.2086063 0.7382779 0.54959095
0.5424881 1.423234   1.4581031 1.034925   0.6970035 0.6646796
1.0944976 0.38276762 1.316532   1.3317639 0.81916595 0.71536213
0.8714377 0.1769578 0.84768665 0.31414166 1.1556908 0.7502223
1.4889848 0.68889725 1.6814903 0.3687363 0.35714096 1.0730665
0.48784363 1.5549862 1.5386477 1.0180881 0.9544302 0.85537875
0.45629108 0.9533397 0.5466732 0.59993136 1.7907507 0.8246471
0.3679787 1.3147795 1.639843   1.0240519 0.938287   1.3132591
1.4715621 1.0977529 1.1388764 1.757766   1.4400833 1.8458658
0.5191678 1.6522799 1.7296914 0.7441294 1.2524359 0.68360096
1.2484676 0.5271824 0.56084824 0.24649574 0.7358151 0.13132821
0.791124   1.136645   1.4944751 0.34966537 0.49364287 1.0588337
1.2211418 0.9757402 0.9296019 1.1600068 1.1461117 0.5922929
0.75524414 0.42948335 1.206921   0.6876476 1.0851916 1.2295783
0.27516115 1.5850624 1.1035944 0.58786446 1.0250506 1.0021445
1.1813903 1.143613   0.7019206 0.46305132 1.0377712 1.627239
0.528895   0.51969105 1.5968494 1.0354733 0.5020237 1.0624642
0.93814045 1.3696482 1.066417   0.5951625 1.6964085 0.8187087
0.99275887 1.1037145 1.2597413 1.2226396 1.4581172 1.2950816

```
0.09459838 0.813329   1.3246593  0.7503948  1.2396047  1.2972338
0.83550537 1.5817835  1.2079396  1.0665135  1.3359131  1.4902492
0.4529717  1.1965088  1.4397542  1.1689749  0.9597262  0.39060745
0.37084436 1.1680717  1.3710408  0.74823356 1.1694151  1.4152086
1.3073117  0.8647845  1.4022815  1.8983977  1.0711551  0.86642134
1.2306218  1.208463   1.0350602  0.723469   0.73401374 1.6717496
0.98212713 0.41451058 0.79827684 1.5695684  1.2458702  1.3649576
1.1521989  0.43884    0.9946438  1.8391811  0.7576993  0.25673324
0.9319821  1.3844243  1.2008996  1.0353287  1.3368082  0.7556171
1.5446451  0.7317848  0.65514207 0.6336172  1.060998   0.58363736
1.3299847  0.7915297  0.06206099 1.1119602  1.1335769  1.313989
0.57426614 1.1290871  1.0171236  1.1443467  0.6155356  1.3715808
0.81681997 1.140823   0.9758166  0.57445985 1.184941   1.4594784
1.5963175  0.3415131  0.65032464 0.82171243 0.91699445 0.46134242
1.7215776  1.1246548  1.3518956  0.10401254 0.4162141  0.77722526
0.5973975  0.90388423 1.0193633  0.56824976 0.97388756 0.933545
1.6781485  1.6182522  1.6260552  0.93511057 0.89688724 0.73232925
1.1014105  1.5928938  0.2604636  1.6984146  1.5488403  0.7069106
0.47085544 1.6244497  1.574141   1.2459021  1.1994077  1.0399132
0.9641064  1.666151   0.690168   1.2830434  0.49647358 0.6738842
1.0830144  0.6864279  0.87099946 0.94649464 0.98832065 1.3161618
0.531778   0.5989059  0.37693134 0.63449466 1.1455078  0.7314316
1.6128697  1.5938832  0.22948003 1.6130414  1.10327    0.9497167
1.0684288  0.89800614 1.4745532  1.8509262  0.45810872 1.865683
0.60974395 0.5349195  1.5187513  0.7561664  0.8521848  0.5181713
1.5662651  1.0417523  1.4160513  0.21841052 1.1247134  0.38147998
0.40628925 0.5688617  1.1104616  1.1202856  1.1840202  0.8467035
1.384716   0.72256684 1.087147   0.23924537 1.0143185  1.4054751
0.8828634  1.6734557  1.3509139  0.64829445 1.3209231  1.2403609
1.1135527  0.3056097  1.0073528  0.64866287 0.8371779  1.521255
1.371338   0.9675938  0.73350674 1.3891538  1.1290506  1.4475904
1.1849611  1.012301   0.47297278 0.677752   1.22894    1.3455629
0.89515877 0.4347314  0.76062894 0.9952374  1.2236123  0.7767668
1.6921065  1.2271156  0.5938357  0.63883746 1.0479223  1.467499
1.4334201  1.0428323  0.8524381  1.7248416  1.2646078  1.4770014
1.1499317  0.47743922 1.056717   1.6417241  0.770213   1.2655758
1.1038351  1.8176614  0.96934617 0.96333337 1.905009   1.178583
0.8065063  0.54421675 1.5981266  1.1952287  0.8542515  0.70768434
0.94853497 1.1805356  0.7466401  1.0538043  0.20293584 1.2322668
1.0352721  1.6740556  0.4365319  0.80784774 1.6198273  1.4734776
1.0942204  0.7003712  1.5993738  1.5829567  1.3238277  0.59076154
1.0509744  0.54625046 1.2514044  0.8360392  0.64322644 1.2077131
0.95344746 0.951218   1.3292601  1.2554835  1.1515288  1.0820816
0.4000591  1.3336054  0.85165787 0.41945782 0.48921937 0.8750268
1.1422141  0.43865925 1.8690562  0.79596615 0.60617155 1.2725632
0.9852519  0.7633526  0.4540338  1.0974354  0.7162572  1.3296261
1.5133204  1.4239846  1.2209405  1.1462368  0.89329576 1.1080523
0.64123094 1.309912   1.7452958  1.2174703  1.6239195  0.60411465
1.392718   1.406269   1.1866457  0.8079778  1.5604086  0.31696486
0.64079016 1.1020939  1.6524465  0.57223713 1.2639349  0.91014636
```

```
1.1953222  1.0311916  1.0282036  1.2278045  1.1570787  0.87157536
1.7275693  0.27073503 1.0724373  0.5258424  1.2086129  1.285188
1.0547878  0.3753549  0.7602218  1.0116477  1.4166894  0.7913173
1.1941189  0.3762981  1.2259121  1.0111886  1.1965483  1.4985151
1.0483278  0.37020487 0.648637   0.4991084  0.484911   1.3017832
0.9651295  0.6372432  1.1473781  0.8787359  0.5869852  0.6177417
1.7162006  0.66583997 0.61997956 0.9262206  1.6281933  0.39115462
0.97601604 1.5089188  1.7663412  1.0978565  0.5749871  1.3285289
1.5987061  1.0389801  0.74245703 0.4343452  0.9901191  0.67714536
1.2186174  0.33659127 1.5530326  0.7086743  1.4868805  0.4070406
1.0774316  1.6369941  1.7237349  1.4996479  1.2886338  0.97823906
0.8787904  0.7081922  1.0385213  1.1342325  1.141727   0.9004595
1.5325968  0.8398268  0.23388124 1.3779991  1.8013074  1.7473221
1.1047894  1.1154835  0.99845135 1.70943    1.5546931  0.5666455
0.9363067  1.6664844  0.66697586 0.91522026 0.35405403 1.6296723
0.7499877  1.5388336  1.3310118  1.0620328  1.5944763  0.6544566
1.0054882  1.267515   0.9196526  0.12913935 1.1490495  1.9015479
0.7423077  1.2083471  0.48859304 1.6466465 ]
```

Importing Libraries:

- `numpy` is imported to work with arrays in Python.
- `numba.cuda` is imported to utilize CUDA for parallel computing.

Kernel Function Definition (`add_arrays`):

- This is a function decorated with `@cuda.jit`, indicating that it's a CUDA kernel.
- It takes three arguments: `a`, `b`, and `c`.
- Inside the kernel function, it computes the element-wise sum of arrays `a` and `b` and stores the result in array `c`.
- `cuda.grid(1)` returns the thread index within a one-dimensional block, which is then used to determine which element of the arrays each thread will process.
- The `if` condition ensures that only threads within the array bounds participate in computation.

Main Function:

- `main()` function initializes arrays `a` and `b` with random float values using NumPy.
- It allocates memory for array `c` on the CPU using `np.zeros_like(a)` which creates an array of zeros with the same shape and type as `a`.
- Memory for arrays `a`, `b`, and `c` is allocated on the GPU using `cuda.to_device()`, which copies the arrays from the CPU to the GPU.
- The kernel function `add_arrays` is launched on the GPU using `cuda.gpus[0]` to select the first GPU. The `add_arrays` kernel is executed with `N` threads, each processing one element of the arrays.
- After the kernel execution, the result array `c` is copied back from the GPU to the CPU using `d_c.copy_to_host(c)`.
- Finally, the result array `c` is printed.

2. Reduction Clause:

```
import numpy as np
from numba import cuda

# Define the kernel function with OpenACC pragma for parallel
execution
@cuda.jit
def sum_reduction(a, result):
    tid = cuda.threadIdx.x
    block_size = cuda.blockDim.x
    grid_size = cuda.gridDim.x
    stride = block_size * grid_size

    # Shared memory for block-level reduction
    shared_sum = cuda.shared.array(256, dtype=np.float32)

    # Perform the reduction within each block
    local_sum = 0
    for i in range(tid, len(a), stride):
        local_sum += a[i]

    # Store the local sum into shared memory
    shared_sum[tid] = local_sum
    cuda.syncthreads()

    # Perform block-level reduction using shared memory
    for s in range(block_size // 2):
        idx = 2 * s * tid
        if idx < block_size:
            shared_sum[idx] += shared_sum[idx + 1]
        cuda.syncthreads()

    # Store the block-level reduction result to global memory
    if tid == 0:
        result[0] = shared_sum[0]

# Main function
def main():
    # Array size
    N = 10000
    # Initialize array with random values
    a = np.random.rand(N).astype(np.float32)

    # Allocate memory on the GPU
    d_a = cuda.to_device(a)
    result = cuda.to_device(np.array([0], dtype=np.float32))

    # Define grid and block dimensions
    block_dim = 256
```

```
    grid_dim = (N + block_dim - 1) // block_dim

    # Launch kernel
    sum_reduction[grid_dim, block_dim](d_a, result)

    # Copy result back to host
    cuda.synchronize()
    final_result = result.copy_to_host()[0]

    print("Sum of array elements:", final_result)

if __name__ == "__main__":
    main()
```

```
Sum of array elements: 43.49064

/usr/local/lib/python3.10/dist-packages/numba/cuda/dispatcher.py:536:
NumbaPerformanceWarning: Grid size 40 will likely result in GPU under-
utilization due to low occupancy.
  warn(NumbaPerformanceWarning(msg))
```

Importing Libraries:

- `numpy` is imported to work with arrays in Python.
- `numba.cuda` is imported to utilize CUDA for parallel computing.

Kernel Function Definition (`sum_reduction`):

- This kernel function performs a sum reduction on an input array `a`.
- It utilizes CUDA for parallel execution.
- It takes two arguments: `a` (the input array) and `result` (an array to store the final result).
- The function is decorated with `@cuda.jit`, indicating it's a CUDA kernel.
- Inside the kernel:
  - It retrieves the thread ID, block size, grid size, and stride.
  - Shared memory `shared_sum` is allocated for block-level reduction.
  - Each thread calculates a local sum within its block.
  - The local sums are stored in shared memory.
  - Block-level reduction is performed using shared memory.
  - The final block-level reduction result is stored in global memory.

Main Function (`main()`):

- The main function initializes the array `a` with random float values.
- Memory for array `a` is allocated on the GPU using `cuda.to_device()`.
- Memory for the result is also allocated on the GPU.
- Grid and block dimensions are defined to launch the kernel with appropriate parallelization.
- The kernel `sum_reduction` is launched on the GPU.
- The result is copied back to the CPU.

- The final result, which is the sum of array elements, is printed.

Execution:

- When executed, this code performs a parallel sum reduction of an array on the GPU using CUDA.
- It utilizes shared memory and block-level reduction to efficiently compute the sum.
- The result is then copied back to the CPU for further processing or display.

3. Loop Optimization:

```
import numpy as np
from numba import cuda

# Define kernel function with loop optimizations
@cuda.jit
def square_elements(a, b):
    idx = cuda.grid(1)
    if idx < len(a):
        # Perform loop optimizations (e.g., loop unrolling)
        sum = 0
        for i in range(10):
            sum += a[idx] * a[idx]
        b[idx] = sum

# Main function
def main():
    # Array size
    N = 1000

    # Initialize input array on the CPU
    a = np.random.rand(N).astype(np.float32)

    # Allocate output array on the CPU
    b = np.zeros_like(a)

    # Allocate memory on the GPU and transfer data
    d_a = cuda.to_device(a)
    d_b = cuda.to_device(b)

    # Launch kernel with data regions and loop optimizations
    with cuda.gpus[0]:
        square_elements[N, 1](d_a, d_b)

    # Transfer result back to the CPU
    d_b.copy_to_host(b)

    # Print result
    print("Result array (b):", b)
```

```
if __name__ == "__main__":
    main()
```

```
Result array (b): [2.82750368e-01 6.51936102e+00 6.88080788e+00
6.67099905e+00
 9.01152611e+00 5.52901983e+00 9.48503399e+00 1.62459326e+00
 1.34035540e+00 3.38841408e-01 2.54158401e+00 6.42568636e+00
 8.11347961e-02 6.53884125e+00 1.23186886e+00 1.88536298e+00
 2.01648474e-02 3.28639960e+00 1.32596266e+00 9.43049622e+00
 3.63169275e-02 2.41000915e+00 1.66276157e+00 8.99422228e-01
 3.10910165e-01 9.41649377e-02 3.11517024e+00 1.87472641e+00
 1.32113855e-05 7.61527634e+00 1.18282318e-01 6.92205317e-03
 2.62890625e+00 2.87200958e-01 9.36566448e+00 7.88177776e+00
 3.51897240e+00 4.12872696e+00 5.47927558e-01 1.83146477e+00
 6.52820396e+00 9.02347453e-03 6.22714832e-02 3.77670586e-01
 6.32136297e+00 7.84441009e-02 6.08687305e+00 3.45068127e-02
 5.08125830e+00 1.60449088e+00 2.26710439e-02 3.59896161e-02
 2.26083064e+00 4.42354298e+00 1.10703254e+00 4.73594379e+00
 1.48112357e+00 1.62149024e+00 4.34323400e-01 5.71614027e+00
 5.88607740e+00 7.48338938e-01 8.52745771e-01 5.66099119e+00
 8.36541939e+00 9.57377052e+00 2.10068369e+00 5.37781954e-01
 4.36520290e+00 1.77136803e+00 8.78990650e+00 2.13002849e+00
 6.66407967e+00 9.89725399e+00 7.97996521e-01 3.37060118e+00
 4.66140461e+00 7.21747589e+00 5.43273878e+00 1.45363450e-01
 7.72656631e+00 7.00667000e+00 6.75224972e+00 8.86959743e+00
 1.14349985e+00 6.19530201e-01 6.77839279e+00 2.18383598e+00
 1.29839778e-01 2.79196358e+00 8.99974108e-02 1.61124796e-01
 1.07134774e-01 7.05953074e+00 3.68396789e-01 5.19510269e-01
 1.44582912e-01 6.14079595e-01 2.35681152e+00 1.53864312e-04
 5.76365376e+00 5.17739415e-01 1.01510930e+00 2.17312837e+00
 5.76130867e+00 2.14326143e+00 5.11822796e+00 6.26586199e+00
 9.94181395e-01 1.17940074e-02 9.12918854e+00 7.07608509e+00
 1.04196227e+00 2.74275589e+00 1.35062803e-02 2.23944616e+00
 7.21122563e-01 3.68464565e+00 3.97848375e-02 8.00014400e+00
 2.07624376e-01 1.42350852e-01 4.25825685e-01 4.56841528e-01
 1.20958351e-01 1.00609922e+00 4.61933708e+00 8.75682259e+00
 2.79963231e+00 4.29182768e+00 6.89915754e-03 1.30623412e+00
 4.93184686e-01 6.61233187e+00 2.33571982e+00 3.01726317e+00
 3.60082197e+00 5.20273972e+00 6.47526979e-01 3.47570157e+00
 4.86274064e-03 3.18462396e+00 3.12191397e-01 4.29945278e+00
 2.77045965e+00 3.62720919e+00 3.36444235e+00 1.55006385e+00
 7.52265692e+00 5.27432479e-04 4.36627626e+00 2.52091466e-03
 3.94291353e+00 1.73858181e-01 4.46755737e-01 5.65846634e+00
 2.23251295e+00 6.54337406e+00 6.81103945e+00 9.51837826e+00
 5.44809923e-02 2.31491709e+00 2.03868604e+00 5.91675091e+00
 3.35844302e+00 4.32438374e+00 9.86339569e+00 2.02563190e+00
 9.20865154e+00 1.18975174e+00 7.02121305e+00 1.94066978e+00
 2.67663687e-01 5.85686028e-01 4.36245203e+00 8.74992087e-03
 4.85604906e+00 5.79280853e-01 3.60738897e+00 8.98315239e+00
 4.73126602e+00 2.42642616e-03 4.97477961e+00 2.10005254e-01
```

```
9.80150223e-01 3.20978522e+00 1.46756279e+00 2.41379929e+00
2.03572139e-01 2.41132045e+00 3.26387167e-01 8.34121418e+00
3.93092111e-02 9.36568677e-02 8.58534431e+00 2.62630510e+00
1.47166580e-01 4.47176886e+00 6.21663332e-01 6.27743721e-01
5.70171833e-01 9.08372688e+00 4.77576447e+00 3.08933997e+00
8.01902890e-01 6.61515951e-01 3.16289926e+00 1.67151940e+00
8.03215206e-01 3.61710024e+00 1.86942887e+00 8.20006752e+00
8.71133804e+00 5.52987218e-01 8.41278744e+00 6.51589304e-04
2.39657052e-02 4.89012384e+00 6.11954880e+00 1.40111613e+00
5.98312902e+00 9.00164127e+00 6.93873835e+00 9.61153567e-01
5.33549534e-03 7.82623243e+00 1.09301186e+00 3.81871128e+00
6.81940460e+00 8.88958931e-01 4.21710300e+00 1.95702299e-01
1.19855332e+00 3.69334500e-03 1.28384352e+00 2.99423170e+00
3.05061865e+00 9.47877884e-01 5.98339272e+00 5.41538102e-05
1.35362849e-01 5.50027072e-01 2.89033890e-01 6.47444963e+00
8.34646702e+00 3.69955492e+00 2.38473833e-01 5.29232979e+00
1.29604411e+00 2.54947758e+00 1.97590232e+00 7.84810925e+00
2.06612802e+00 1.37522280e-01 1.22073062e-01 7.41953373e+00
4.43894815e+00 1.14473470e-01 2.95185041e+00 1.87031962e-02
4.87208217e-02 4.69286966e+00 9.49838221e-01 1.73640060e+00
1.07377732e+00 7.80624211e-01 7.56646916e-02 7.98010635e+00
2.58147764e+00 4.07726049e+00 1.87069964e+00 7.36429501e+00
1.93101525e-01 1.04677202e-02 9.54940510e+00 1.17823553e+00
3.33352661e+00 3.20782065e-01 2.64016867e+00 2.85302019e+00
2.79310765e-03 3.79846907e+00 9.90490377e-01 4.13186455e+00
3.71048903e+00 1.03765585e-01 9.05736065e+00 6.06266737e+00
3.26612115e+00 1.25988650e+00 8.76771584e-02 8.60397434e+00
1.61340368e+00 4.96469307e+00 4.09927934e-01 2.55945563e+00
9.73538220e-01 3.92980552e+00 3.45271975e-02 2.56714082e+00
6.59010351e-01 3.80856538e+00 7.54529047e+00 8.20620716e-01
3.91684741e-01 7.30551779e-01 1.08490251e-02 3.88092732e+00
9.92315388e+00 3.09121609e+00 3.13792276e+00 7.74161518e-01
3.00747299e+00 7.04903364e+00 1.09184480e+00 6.74785376e+00
2.05491376e+00 1.39895350e-01 2.42564589e-01 5.19579470e-01
7.04470444e+00 9.16314220e+00 4.66285801e+00 6.46615624e-01
2.25772357e+00 7.38361168e+00 1.28849208e+00 8.90629673e+00
6.10853612e-01 5.74562502e+00 8.78828144e+00 4.57997751e+00
2.99000239e+00 3.47342372e-01 6.28952885e+00 1.50361466e+00
8.07635689e+00 2.74515843e+00 7.60642147e+00 2.52850509e+00
1.82677820e-01 3.48226726e-01 5.20891953e+00 3.73927981e-01
2.32640756e-04 6.68084383e+00 2.65826178e+00 4.70758629e+00
1.07749200e+00 5.88551521e-01 9.48171377e-01 4.17989540e+00
7.69243431e+00 3.31908178e+00 4.46997941e-01 4.72413921e+00
2.68429041e+00 9.16591263e+00 1.72514826e-01 8.64288235e+00
3.16756725e+00 3.22243214e-01 3.59824240e-01 4.03819196e-02
5.35297394e-01 2.05932212e+00 6.54332066e+00 7.28565502e+00
4.67472935e+00 9.06060505e+00 9.18206692e+00 2.25894403e+00
1.11470167e-02 4.10869551e+00 8.67461205e+00 2.51110888e+00
6.61116326e-04 2.58444333e+00 3.18043375e+00 7.82529637e-03
```

9.17407894e+00 3.18586111e-01 5.05498457e+00 1.10730255e+00
6.68416858e-01 2.57519507e+00 6.81676626e-01 6.00951537e-02
2.12827134e+00 6.70571804e+00 8.44004631e-01 3.68683338e-01
5.87982655e+00 7.12763548e+00 1.53821751e-01 1.65778303e+00
1.55503201e+00 7.15518379e+00 3.33381438e+00 1.00892246e-01
6.72108126e+00 2.34971666e+00 6.44189358e+00 8.33089650e-01
3.97017241e+00 2.36842752e-01 2.39204836e+00 5.17774343e+00
1.77574527e+00 6.20841980e+00 1.17141461e+00 2.28454709e-01
3.13406531e-03 9.86023521e+00 8.13130951e+00 2.04961276e+00
3.52687216e+00 2.09814325e-01 9.82331038e-01 2.97181988e+00
1.63231060e-01 1.21862844e-01 1.21348418e-01 3.42406726e+00
7.17042160e+00 7.74042177e+00 2.87722158e+00 6.14265394e+00
3.73698831e+00 2.07967430e-01 1.40669540e-01 1.21517885e+00
8.65893513e-02 5.84362540e-03 4.88919544e+00 7.28943825e+00
9.66122150e-01 5.37647152e+00 8.15119222e-02 5.21784353e+00
5.87845683e-01 5.32650054e-02 1.47194505e-01 9.45799798e-03
1.96211450e-02 5.31903505e-02 4.95108509e+00 7.05440235e+00
9.26730824e+00 2.44657367e-01 3.16936827e+00 1.84484673e+00
1.88080537e+00 5.61688709e+00 3.23683023e-01 3.19298077e+00
3.96041822e+00 3.11807871e+00 1.60996938e+00 1.62524104e+00
6.20415783e+00 7.39089394e+00 1.87091219e+00 4.58240323e-02
6.02842867e-02 7.77581453e+00 6.57969952e+00 6.79606676e-01
7.67915606e-01 6.26610443e-02 1.98114693e-01 5.47296703e-02
3.78848362e+00 1.07732344e+00 2.76868248e+00 8.54768944e+00
4.19347221e-03 7.55833673e+00 8.51166534e+00 3.68608928e+00
3.64873976e-01 7.61326075e+00 3.96773553e+00 1.15303390e-01
3.76531512e-01 9.86041355e+00 1.90243912e+00 1.78624094e+00
1.43563819e+00 2.15403080e+00 1.22580957e+00 1.15931070e+00
1.96729135e-02 5.83173573e-01 6.63466454e+00 4.36722183e+00
5.53821373e+00 3.55993772e+00 1.08019665e-01 9.54247832e-01
1.21746886e+00 1.30552202e-01 3.52941942e+00 8.02743137e-01
6.71066332e+00 1.57163575e-01 2.22079322e-01 1.83261544e-01
8.73513317e+00 9.17126526e-05 1.83384903e-02 4.32189226e-01
2.51118135e+00 7.66993403e-01 1.89349365e+00 4.35949415e-02
6.06699705e-01 3.11872983e+00 8.07956982e+00 1.78108466e+00
3.95731568e-01 1.03021264e+00 3.52162600e+00 5.12644928e-03
7.16436505e-02 2.84489878e-02 2.04912996e+00 3.54385948e+00
7.57739902e-01 3.88882756e+00 2.97151245e-02 3.57709789e+00
5.78887558e+00 8.02601910e+00 2.80641586e-01 4.33240223e+00
5.19271660e+00 8.23661423e+00 1.86391258e+00 7.88817763e-01
5.97406149e+00 7.22569180e+00 5.68704653e+00 1.16472691e-02
3.67528766e-01 1.22155809e+00 6.49188638e-01 4.29991817e+00
4.62758923e+00 4.11524153e+00 8.72995663e+00 7.46973419e+00
3.76724422e-01 9.31414892e-04 8.48920631e+00 3.86568093e+00
1.07512772e-01 2.79045391e+00 2.11116195e-01 3.18663788e+00
6.59069180e-01 1.47095025e+00 1.77485895e+00 2.27055573e+00
9.44338977e-01 4.56933454e-02 5.49057436e+00 2.80399919e+00
1.22567821e+00 5.15673065e+00 9.32163048e+00 6.01324499e-01
8.84902477e+00 4.18342257e+00 8.05959225e-01 9.42847824e+00

7.62572098e+00 8.43885612e+00 4.35859776e+00 5.05827665e+00
1.04118988e-01 2.94992566e+00 1.21311510e+00 1.41903901e+00
3.99627185e+00 1.40310198e-01 2.88875371e-01 7.25993454e-01
3.05527687e-01 2.96366405e+00 2.82369876e+00 3.44801712e+00
3.83684373e+00 4.10671711e+00 5.83908856e-01 5.39755392e+00
3.40157866e+00 1.62693262e+00 2.73504448e+00 7.24522948e-01
1.98177862e+00 7.51041842e+00 3.37883830e-01 2.31063867e+00
4.01725101e+00 3.60224771e+00 1.03820185e-03 6.21608210e+00
2.26236701e+00 8.78411674e+00 2.27893972e+00 7.09937871e-01
1.53854012e+00 1.25428820e+00 4.17252588e+00 1.55402213e-01
8.27085257e-01 9.40545273e+00 5.89682698e-01 6.19680166e+00
1.68152833e+00 8.87021124e-01 1.29647481e+00 8.37251842e-01
5.96375883e-01 4.04303932e+00 1.17920637e+00 1.36806607e+00
1.97776175e+00 7.93005228e+00 4.74255371e+00 9.60667253e-01
8.15844917e+00 1.78408604e-02 9.30293941e+00 9.76939946e-02
1.88385582e+00 1.36633921e+00 8.81367247e-04 1.22423479e-02
4.04179049e+00 1.16069448e+00 1.77869555e-02 4.49631596e+00
4.35560048e-02 5.46022117e-01 4.26609468e+00 8.01068592e+00
9.33495617e+00 4.76182032e+00 7.94770765e+00 2.74556689e-02
5.31293809e-01 5.69380999e-01 2.16445494e+00 5.09024239e+00
7.33690643e+00 2.52248999e-03 2.77472448e+00 1.90375671e-01
3.89996719e+00 3.62643504e+00 9.85505199e+00 1.09846741e-01
7.37590837e+00 6.57137632e+00 5.14573956e+00 2.11220169e+00
2.22889805e+00 2.73817754e+00 8.53066802e-01 6.61417842e-01
4.28599358e+00 5.62469053e+00 1.91848564e+00 4.71836519e+00
8.60450625e-01 4.39697838e+00 8.00816345e+00 2.35454464e+00
7.58078337e+00 4.41594869e-01 1.83774805e+00 1.26621991e-01
5.62820864e+00 5.30376911e+00 4.25538731e+00 8.13532257e+00
1.10498810e+00 6.90400887e+00 3.09090185e+00 2.69559473e-01
7.67528105e+00 3.26426148e+00 8.86431217e-01 2.59682536e-01
3.44272563e-03 5.28266811e+00 1.77648199e+00 1.56296468e+00
8.02303219e+00 9.78423119e+00 7.30476284e+00 6.72212553e+00
9.51765251e+00 4.81658816e-01 5.31875372e+00 7.76218319e+00
5.25883102e+00 3.45217586e+00 7.39382267e+00 4.94422150e+00
1.74110651e+00 5.98610449e+00 2.82382488e+00 6.32004929e+00
7.62325749e-02 6.83673811e+00 2.50912142e+00 9.62904263e+00
9.50085735e+00 2.82274866e+00 4.82662344e+00 5.75583220e-01
3.91562009e+00 1.62551969e-01 1.78159833e+00 7.19441652e-01
3.32086205e+00 8.42619538e-01 7.77603388e+00 6.28812790e-01
1.70304513e+00 2.39919901e+00 2.19985628e+00 6.64810133e+00
4.41558981e+00 5.30328304e-02 1.14252353e+00 3.70489836e+00
3.06420183e+00 9.29791832e+00 1.39944053e+00 7.93681443e-02
1.39879441e+00 3.08963299e-01 4.65848637e+00 6.87766743e+00
5.34931421e+00 2.21278167e+00 8.79691792e+00 6.11760426e+00
3.89862359e-02 7.45176411e+00 2.79077268e+00 4.53259182e+00
3.26242304e+00 2.18254280e+00 1.31134808e-01 6.85503602e-01
3.08538854e-01 1.51302767e+00 6.57320786e+00 4.71237278e+00
4.53964520e+00 5.46186352e+00 4.56990767e+00 7.51530647e+00
4.33592701e+00 4.69762611e+00 5.25272369e+00 5.17620516e+00

2.89680719e-01 7.61067104e+00 1.71949342e-01 1.62981912e-01
2.79132748e+00 2.07050514e+00 3.04563314e-01 1.37032747e-01
8.99219513e-01 3.62217712e+00 6.51201916e+00 8.06098843e+00
4.01090240e+00 5.69934559e+00 1.09696853e+00 1.28410859e-02
8.11563396e+00 2.62687159e+00 1.95936871e+00 6.93144798e-01
1.73791158e+00 2.09866118e+00 1.80234444e+00 2.13629413e+00
9.42118108e-01 2.71247268e-01 4.18928337e+00 7.90860128e+00
1.31609631e+00 4.42304230e+00 3.27965856e+00 1.17605472e+00
2.15642489e-04 4.11049277e-01 4.55333702e-02 5.10219097e-01
4.73220634e+00 5.27674866e+00 4.30908985e-03 2.43190583e-02
4.19791842e+00 6.13765442e-05 1.69425178e+00 4.07434988e+00
8.96799850e+00 5.37670612e+00 1.69108827e-02 4.20618916e+00
3.20555806e+00 6.00193310e+00 2.61945510e+00 8.81740689e-01
3.61509800e+00 4.42340708e+00 9.42401171e-01 1.76931596e+00
2.71009111e+00 7.86152363e+00 5.59564495e+00 5.63378239e+00
2.39126015e+00 6.17057705e+00 9.51784897e+00 1.41472235e-01
6.25581264e-01 7.01800919e+00 9.31589890e+00 1.86217260e+00
8.25695896e+00 7.82194436e-01 2.71895123e+00 1.02349913e+00
2.67437410e+00 4.59547091e+00 1.10825729e+00 1.84605551e+00
9.09658447e-02 6.64179206e-01 3.36002207e+00 3.16610664e-01
7.75058493e-02 6.95874393e-02 7.87764549e+00 2.73762035e+00
1.82316768e+00 3.78953576e+00 3.91921341e-01 5.68047905e+00
1.47778678e+00 6.32317543e+00 1.58706200e+00 2.84945339e-01
1.92312822e-01 7.20980883e-01 4.39628929e-01 2.11959743e+00
7.70744228e+00 2.06217003e+00 8.53775516e-02 3.69267607e+00
6.15669298e+00 8.11813116e-01 7.15272713e+00 4.45621014e-01
8.68359149e-01 9.42314434e+00 9.85775852e+00 8.75917053e+00
7.55916178e-01 1.87978595e-01 7.06673265e-02 4.52305508e+00
2.42212045e-04 5.15750980e+00 4.34210110e+00 4.10163850e-02
4.57604456e+00 1.99216534e-03 8.93198872e+00 4.58424187e+00
3.01625013e+00 1.33945763e-01 4.21486825e-01 2.50394678e+00
4.31430054e+00 6.94244576e+00 1.94789901e-01 3.96710300e+00
5.42342377e+00 5.71605563e-01 3.15516621e-01 1.36199820e+00
5.33079195e+00 5.66097689e+00 3.51377606e+00 2.04474255e-01
6.94018888e+00 3.80611992e+00 5.18305874e+00 3.71061325e+00
3.21406312e-02 1.36529595e-01 1.58667043e-01 7.80130643e-03
2.16646528e+00 4.33963346e+00 7.42249250e+00 1.20058131e+00
2.14498401e+00 3.11789703e+00 4.35399008e+00 4.42461312e-01
1.23857391e+00 1.98736739e+00 6.67622685e-03 4.98463400e-03
1.20485516e-03 5.47563314e+00 1.65738750e+00 2.20426247e-02
5.22737980e-01 3.74739981e+00 1.29733467e+00 1.36726588e-01
3.16449857e+00 2.20631771e-02 4.31085765e-01 9.43377893e-03
3.58036137e+00 1.59264994e+00 7.16830826e+00 3.86590213e-01
2.36094165e+00 4.63575171e-03 6.73058510e+00 1.65777421e+00
6.21892357e+00 8.47197175e-02 4.86472559e+00 1.56404555e-01
2.77289525e-02 5.19823265e+00 6.66178608e+00 5.91728640e+00
9.50666809e+00 3.23616624e-01 6.89242649e+00 9.82375526e+00
6.21270180e-01 5.20677519e+00 2.50771332e+00 2.74837041e+00
6.01179488e-02 2.20209169e+00 2.55742979e+00 7.40346014e-01

```
5.42832553e-01 1.09192654e-01 2.72768974e+00 3.79103613e+00
7.58123100e-01 7.85202626e-03 4.14723349e+00 1.14529625e-01
8.88054752e+00 3.36894058e-02 2.18628430e+00 3.68141747e+00
7.13950396e+00 8.80769825e+00 3.58714461e-01 7.48111629e+00
1.62838727e-01 2.15915251e+00 8.33679922e-03 2.51300001e+00
7.29451716e-01 9.47974396e+00 7.59202540e-01 3.47732353e+00
7.89116472e-02 4.65261316e+00 9.66340542e-01 6.65272951e+00
6.11192048e-01 2.24775672e+00 1.14469182e+00 1.07476151e+00]
```

Importing Libraries:

- `numpy` is imported to work with arrays in Python.
- `numba.cuda` is imported to utilize CUDA for parallel computing.

Kernel Function Definition (`square_elements`):

- This kernel function calculates the square of each element in the input array `a` and stores the result in array `b`.
- It utilizes CUDA for parallel execution.
- It is decorated with `@cuda.jit`, indicating it's a CUDA kernel.
- Inside the kernel:
  - `cuda.grid(1)` is used to retrieve the thread index within a one-dimensional block.
  - Each thread calculates the square of the corresponding element in array `a`.
  - The result is stored in the corresponding position in array `b`.
  - Loop optimizations, such as loop unrolling, are applied for better performance.

Main Function (`main()`):

- The main function initializes the input array `a` with random float values.
- Memory for the output array `b` is allocated on the CPU using `np.zeros_like(a)`.
- Memory for arrays `a` and `b` is allocated on the GPU using `cuda.to_device()`.
- The kernel `square_elements` is launched on the GPU.
- The result is copied back to the CPU.
- The resulting array `b` containing the squares of elements is printed.

Execution:

- When executed, this code performs parallel computation of the squares of elements in an array on the GPU using CUDA.
- Loop optimizations are applied within the kernel for better performance.
- The result is then copied back to the CPU for further processing or display.

4. Parallel and kernel directives:

```
import numpy as np
from numba import cuda

# Define kernel function
```

```python
@cuda.jit
def multiply_elements(a, b, c):
    idx = cuda.grid(1)
    if idx < len(a):
        c[idx] = a[idx] * b[idx]

# Main function
def main():
    # Array size
    N = 1000

    # Initialize input arrays on the CPU
    a = np.random.rand(N).astype(np.float32)
    b = np.random.rand(N).astype(np.float32)

    # Allocate output array on the CPU
    c = np.zeros_like(a)

    # Allocate memory on the GPU
    d_a = cuda.to_device(a)
    d_b = cuda.to_device(b)
    d_c = cuda.to_device(c)

    # Launch kernel with parallel directive
    block_size = 256
    grid_size = (N + block_size - 1) // block_size

    multiply_elements[grid_size, block_size](d_a, d_b, d_c)

    # Transfer result back to the CPU
    d_c.copy_to_host(c)

    # Print result
    print("Result array (c):", c)

if __name__ == "__main__":
    main()
```

```
Result array (c): [2.02094227e-01 9.43410993e-02 3.86913896e-01
1.29064202e-01
 3.20593596e-01 1.97348729e-01 5.85442558e-02 4.25929189e-01
 1.12647630e-01 1.34312719e-01 1.18314624e-01 2.98026875e-02
 4.59079415e-01 1.61380306e-01 8.54636490e-01 6.75200969e-02
 2.06291750e-01 3.04460436e-01 2.11886689e-01 2.86172032e-01
 1.43993109e-01 2.71580607e-01 8.72256100e-01 1.35649502e-01
 4.62028027e-01 4.29785103e-02 3.25537324e-01 3.47620100e-02
 7.97379836e-02 3.81804496e-01 2.35465318e-01 1.07487142e-01
 1.49909809e-01 9.51573700e-02 1.49219945e-01 1.07929312e-01
 2.25038137e-02 1.93623245e-01 4.25525963e-01 1.02540612e-01
 2.48345688e-01 1.74852088e-01 1.65970298e-03 3.24685693e-01
```

6.16362274e-01 5.21901786e-01 7.65474975e-01 1.69166595e-01
7.24604607e-01 1.88321061e-02 2.49669120e-01 9.35589373e-02
3.98936689e-01 3.46934438e-01 1.38679564e-01 2.62115672e-02
1.49742872e-01 1.71001013e-02 1.17050640e-01 7.26336613e-03
1.96498990e-01 1.78179401e-03 1.03543773e-01 1.06195807e-01
5.13587177e-01 2.00532779e-01 1.35903642e-01 3.95599613e-03
6.88231215e-02 6.63115904e-02 5.36697209e-01 9.22493413e-02
2.58937627e-01 1.87501818e-01 1.40911788e-02 1.81862429e-01
6.11325055e-02 4.26226556e-02 9.96370986e-03 9.56060663e-02
4.09517825e-01 1.72893405e-01 1.70894995e-01 5.15526533e-02
3.91221605e-02 2.01770037e-01 2.94806451e-01 3.75559300e-01
7.15054035e-01 7.76236318e-03 1.74052864e-01 4.28766340e-01
3.09839249e-01 1.91634323e-03 2.01485112e-01 3.98206621e-01
8.07518482e-01 2.16203243e-01 4.75945929e-03 2.69646049e-01
3.77055109e-01 3.93568009e-01 4.37980238e-03 2.56388158e-01
8.20973963e-02 9.97770727e-02 2.44318292e-01 8.27428639e-01
3.27957839e-01 1.50678262e-01 2.95163602e-01 3.33593816e-01
4.44877803e-01 1.16570145e-02 2.50566185e-01 1.88383877e-01
2.72084743e-01 3.72058243e-01 6.20189607e-02 2.94052232e-02
3.27014253e-02 3.25508118e-01 1.43494578e-02 3.87676984e-01
1.80377588e-01 6.28881037e-01 3.19859177e-01 1.75599176e-02
7.29359090e-01 1.09002069e-01 7.01210618e-01 1.19243469e-02
6.11100614e-01 1.31382972e-01 2.45154306e-01 4.43847537e-01
8.20395425e-02 1.97354965e-02 6.92558885e-01 2.12491140e-01
3.75134677e-01 4.00261104e-01 6.36444330e-01 1.10702321e-03
9.17985797e-01 2.38863289e-01 1.59419738e-02 2.41124764e-01
8.87616351e-02 1.24072701e-01 2.64206260e-01 6.26794219e-01
3.56546402e-01 5.11268497e-01 9.78968292e-02 2.38457341e-02
6.29531562e-01 3.91015738e-01 6.98498070e-01 1.83952272e-01
2.43062619e-02 1.84835762e-01 1.24893531e-01 1.79147467e-01
2.14107096e-01 3.65752690e-02 1.90680206e-01 1.49988204e-01
4.99706477e-01 1.61210045e-01 2.21799657e-01 4.80324514e-02
7.78986886e-02 1.13274157e-01 3.93050343e-01 4.20093328e-01
1.71517015e-01 8.29521343e-02 3.52045655e-01 7.27717340e-01
4.94629443e-01 2.39941329e-01 2.59127259e-01 2.12370113e-01
3.42039287e-01 7.97215849e-04 5.63060120e-02 3.30286354e-01
7.77672045e-03 1.36430636e-02 1.67439327e-01 1.08190827e-01
1.05799772e-02 1.74267843e-01 1.91202313e-01 6.53407395e-01
1.03594914e-01 4.15928721e-01 2.65017822e-02 4.43131924e-01
5.43565869e-01 6.76877916e-01 2.25197926e-01 1.51739553e-01
2.84613837e-02 4.45414722e-01 1.87455621e-02 1.19133748e-01
1.13373779e-01 2.11935993e-02 2.68500578e-02 1.86514884e-01
8.69137526e-01 1.43569812e-01 4.86469179e-01 2.30240703e-01
1.61281899e-01 1.80498734e-01 3.15005243e-01 1.49915358e-02
7.67806843e-02 1.63710073e-01 1.73041923e-03 1.10304974e-01
4.94808435e-01 8.70479226e-01 1.06871314e-01 5.74782789e-01
6.90683305e-01 2.01690704e-01 2.17228770e-01 5.14209330e-01
1.85855702e-01 6.57411039e-01 1.47306457e-01 2.84243841e-02
2.09282219e-01 1.79632159e-03 1.31585360e-01 2.73265243e-01

1.72364175e-01 5.29864252e-01 1.95321470e-01 8.95251632e-02
9.54860542e-03 1.13106705e-02 7.43392706e-02 1.54075295e-01
2.17731789e-01 7.72801161e-01 2.49791339e-01 8.90140086e-02
1.23462509e-02 3.27090290e-03 6.04330450e-02 7.58073351e-04
1.82879150e-01 3.86725187e-01 1.92269906e-01 6.07106149e-01
1.79718480e-01 2.51016676e-01 4.84374911e-01 7.45536610e-02
2.04454362e-01 5.75018048e-01 5.78624234e-02 2.78882802e-01
1.90806121e-01 5.34419231e-02 2.10853722e-02 2.42591262e-01
2.70567179e-01 2.48858929e-01 5.98579288e-01 4.14270103e-01
2.45306604e-02 3.88937771e-01 6.36559427e-02 1.57268956e-01
1.32638380e-01 5.40453076e-01 2.21945897e-01 1.91739216e-01
8.44001114e-01 1.47530399e-02 5.42646348e-01 6.83799610e-02
2.43104100e-01 2.00138614e-01 2.15097502e-01 4.90209103e-01
7.61261582e-02 3.94692242e-01 4.75604892e-01 4.51976880e-02
3.44907850e-01 3.73541564e-01 1.58922188e-02 5.51152050e-01
1.89274754e-02 2.30763927e-01 1.25038341e-01 6.46500170e-01
6.68576546e-03 4.16485280e-01 2.25371927e-01 2.48176664e-01
1.11030541e-01 3.50474343e-02 2.78827727e-01 1.63687110e-01
2.10153341e-01 4.93789762e-02 2.55531762e-02 6.33734837e-02
4.94281083e-01 1.27336923e-02 3.34403813e-01 8.67740884e-02
3.41266207e-02 2.69297808e-01 1.57902896e-01 3.09392452e-01
2.72709946e-03 3.14686209e-01 9.24034059e-01 1.58441499e-01
1.31935954e-01 2.37447228e-02 4.22255211e-02 7.96242133e-02
6.73170993e-03 9.27798171e-03 3.05112720e-01 2.49340221e-01
8.50405335e-01 7.29714930e-01 8.64919350e-02 5.38984954e-01
3.22473943e-01 6.27202615e-02 1.76098794e-01 4.35656488e-01
1.20601453e-01 4.98267174e-01 7.18290150e-01 2.90413387e-02
1.70258880e-01 5.50699793e-02 2.43501678e-01 6.51479065e-01
2.23194975e-02 2.46321276e-01 8.76657013e-03 6.52742863e-01
1.63661703e-01 4.48318422e-01 6.24041967e-02 2.02121288e-02
2.76149452e-01 5.12715876e-01 5.31881213e-01 3.65383849e-02
3.39540809e-01 5.74051857e-01 4.45619524e-02 2.69338578e-01
3.97108942e-02 4.25440520e-02 1.25061303e-01 6.62744105e-01
4.51876730e-01 4.22662914e-01 1.09205030e-01 4.19383734e-01
1.27330437e-01 4.17420208e-01 2.85014778e-01 5.87734953e-02
4.46599498e-02 1.54370204e-01 1.49721280e-01 4.03542221e-02
8.68256688e-02 7.30271101e-01 5.72310984e-01 1.03407718e-01
2.57966191e-01 7.52276301e-01 4.07844603e-01 1.98342223e-02
5.69042563e-02 4.91950572e-01 5.50120473e-01 1.98323876e-01
3.01372223e-02 3.23908925e-01 8.67960602e-02 2.30959989e-02
3.83311599e-01 5.26155171e-04 2.47551620e-01 2.28866220e-01
1.93872359e-02 2.22653747e-01 3.33560854e-02 2.28547379e-01
1.47177324e-01 2.74448335e-01 1.49913996e-01 5.71087837e-01
2.31187537e-01 1.71554655e-01 4.93607391e-03 4.05800074e-01
3.03988844e-01 2.23002031e-01 9.52353999e-02 1.36564940e-01
4.54343468e-01 2.90186435e-01 1.61606476e-01 2.73545712e-01
1.80818990e-01 2.70125687e-01 1.89399540e-01 1.47775665e-01
4.40913707e-01 1.97056308e-01 2.74682015e-01 4.44355011e-02
4.82706636e-01 1.92180514e-01 4.05487940e-02 2.41513923e-01

2.19166628e-03 8.58039930e-02 2.94268429e-01 3.97921726e-02
1.78485572e-01 1.00108922e-01 1.75578073e-02 1.45793110e-01
2.68750787e-01 6.83883950e-02 2.87989050e-01 7.69379586e-02
5.71171977e-02 5.93674898e-01 9.50924233e-02 2.82373160e-01
5.63729823e-01 2.67052561e-01 3.35951447e-01 9.04290974e-01
1.87242061e-01 1.52569577e-01 1.41361365e-02 5.05016088e-01
4.80771996e-03 1.83034867e-01 9.74870503e-01 2.93293566e-01
1.81450501e-01 2.85610855e-01 2.03390896e-01 1.30835965e-01
9.71836131e-03 2.56603032e-01 1.07134841e-01 8.83613348e-01
1.29429966e-01 8.36435929e-02 5.02961129e-03 2.13025451e-01
1.85142606e-02 7.19315708e-02 6.12728655e-01 1.20423166e-02
1.44001991e-01 5.88894308e-01 1.35429710e-01 8.41218606e-02
3.52444202e-01 2.44249627e-02 2.74223506e-01 3.08550056e-02
9.45858210e-02 2.95618057e-01 4.40158188e-01 5.92833817e-01
2.54841596e-01 3.83811355e-01 8.56197178e-02 3.31605598e-02
1.03475012e-01 1.01870030e-01 1.45128705e-02 7.31423050e-02
2.14686990e-01 2.90573388e-01 4.41888385e-02 3.68031055e-01
6.52588248e-01 5.73059879e-02 2.86425531e-01 7.55353644e-02
5.47259636e-02 1.45097286e-01 2.93182373e-01 1.82709303e-02
3.42379719e-01 2.95035243e-02 1.83834601e-02 6.24603331e-01
7.27906287e-01 1.99401215e-01 4.28985506e-02 2.50858158e-01
3.06893438e-01 1.28594413e-01 1.22806497e-01 4.49246287e-01
5.04067019e-02 4.64748405e-02 1.35524407e-01 6.86909109e-02
1.57191768e-01 9.91584539e-01 3.82316500e-01 3.44579548e-01
1.67197153e-01 6.10460162e-01 4.44734842e-01 5.81320748e-03
3.99923831e-01 2.00838581e-01 7.03686833e-01 1.81672946e-01
2.20744405e-03 5.06678343e-01 1.74696103e-01 3.95177770e-03
2.91697055e-01 3.71991664e-01 5.24993651e-02 6.39110863e-01
3.52984428e-01 3.75606678e-02 1.47604913e-01 4.65340883e-01
4.21594828e-01 3.38031590e-01 1.33361518e-01 2.44366720e-01
4.77010518e-01 3.89515638e-01 1.89469904e-01 1.43926428e-03
4.45905179e-02 2.94717222e-01 1.68285877e-01 2.81147957e-02
5.21969855e-01 4.50751811e-01 1.00788809e-01 4.42477614e-02
3.80434334e-01 1.04045004e-01 2.18634501e-01 8.64329398e-01
1.61824077e-01 1.79460887e-02 1.86915532e-01 2.21216306e-02
6.57675803e-01 6.24895453e-01 5.91722243e-02 1.17343977e-01
6.82574153e-01 4.45321985e-02 1.07334517e-01 1.13900611e-02
3.02743372e-02 4.03531730e-01 5.15165806e-01 2.12718651e-01
3.44446272e-01 5.50886393e-01 1.80486534e-02 4.77821082e-02
1.97145671e-01 7.32215941e-01 6.94629997e-02 5.70593655e-01
6.56930311e-03 1.41574100e-01 1.21648796e-02 1.48360044e-01
1.11770049e-01 4.34204489e-02 9.38611478e-02 1.20621383e-01
1.20801121e-01 1.06850192e-01 1.01067320e-01 1.54342487e-01
9.23971646e-04 2.49496456e-02 1.60958409e-01 1.49604842e-01
1.76976640e-02 4.42161085e-03 1.24279559e-01 1.92273618e-03
6.76406980e-01 3.52310807e-01 4.03084248e-01 2.11302832e-01
2.06829794e-02 8.22452009e-02 9.67533827e-01 2.49285638e-01
4.90041971e-02 4.64776546e-01 4.18265849e-01 1.13679074e-01
3.58505547e-01 1.82128127e-03 3.86317641e-01 5.55312634e-01

1.88719660e-01 1.20847814e-01 1.20780647e-01 2.28453070e-01
8.69911134e-01 1.94559582e-02 2.26260811e-01 6.74142301e-01
3.35738599e-01 8.04486964e-03 1.47353396e-01 2.37391330e-02
1.40485615e-01 4.35539782e-01 3.03300977e-01 2.98426926e-01
2.20879555e-01 5.23641169e-01 1.44187614e-01 1.32192850e-01
2.92819180e-02 4.63340908e-01 1.50582209e-01 7.08847880e-01
4.77381051e-01 6.96161315e-02 3.58861923e-01 2.18493342e-01
1.29991740e-01 4.52411324e-02 2.63456464e-01 4.83475596e-04
5.75967021e-02 1.24853849e-01 6.79097548e-02 5.96223585e-02
1.24148197e-01 5.33330590e-02 2.94465609e-02 2.92763621e-01
3.01906496e-01 6.05393872e-02 1.81248099e-01 5.07580042e-01
1.26772836e-01 6.00687444e-01 8.29974934e-02 1.46148652e-01
4.60897014e-02 2.02015206e-01 6.17655367e-02 4.50096637e-01
2.77720809e-01 4.01018821e-02 2.71958143e-01 7.20926046e-01
4.15486358e-02 1.49642187e-03 5.78285940e-02 3.35026920e-01
2.36270994e-01 2.33668252e-03 4.64024067e-01 1.13981940e-01
1.04805902e-02 5.70225045e-02 5.66199087e-02 4.74293321e-01
6.47755666e-03 1.92782119e-01 2.06118777e-01 2.51384586e-01
2.34153852e-01 3.61327715e-02 2.86520541e-01 5.33197671e-02
1.14507347e-01 3.10293376e-01 7.29617476e-01 4.38736640e-02
2.70960778e-01 6.52577877e-01 6.13887748e-03 1.71667948e-01
2.30771258e-01 9.08167735e-02 3.02862734e-01 4.94178027e-01
2.46066019e-01 5.57802558e-01 6.58208802e-02 6.24608099e-01
1.50435045e-02 3.06978464e-01 5.81690490e-01 5.52951843e-02
1.99646413e-01 1.29529357e-01 2.01297272e-02 2.17439085e-01
1.49335429e-01 9.61209163e-02 4.14344639e-01 1.79763827e-02
5.31378016e-02 3.64519149e-01 3.56383055e-01 6.30597413e-01
3.47007990e-01 7.50686228e-02 4.16239887e-01 1.74851105e-01
1.22964971e-01 1.86503440e-01 6.17444441e-02 3.13846059e-02
7.21916378e-01 5.82514592e-02 2.80220271e-03 3.81953269e-02
3.24078113e-01 7.71872178e-02 5.56882331e-03 1.29609033e-01
6.19509481e-02 3.85970116e-01 5.07794693e-02 9.15000811e-02
4.40668970e-01 4.00123090e-01 6.89558327e-01 4.79221910e-01
1.23108618e-01 6.43882811e-01 2.07026139e-01 4.21373814e-01
2.58640915e-01 1.36230335e-01 1.69797927e-01 4.15604264e-02
1.90768808e-01 7.89995864e-02 1.03174821e-02 3.56791437e-01
4.86134738e-02 3.28749716e-01 8.16095099e-02 1.27201408e-01
4.77243476e-02 4.00416963e-02 6.31053925e-01 2.22416315e-02
2.22887740e-01 9.96568240e-03 2.31950268e-01 3.61220181e-01
8.11250985e-01 1.19171157e-01 2.18340382e-01 3.92366275e-02
9.75202695e-02 4.84274402e-02 8.53728294e-01 6.92874938e-02
2.02639848e-01 3.28004897e-01 6.05508164e-02 2.68554781e-02
1.60263807e-01 2.30799675e-01 7.38451719e-01 5.01552343e-01
6.13101959e-01 5.06077707e-02 3.60660404e-01 4.99816477e-01
1.40764803e-01 3.13893259e-01 3.74472827e-01 2.41043895e-01
2.07548246e-01 2.71101091e-02 1.08100593e-01 3.73359233e-01
9.74367112e-02 5.65019667e-01 5.26772916e-01 8.18739310e-02
4.28569913e-02 5.35598814e-01 5.03149629e-01 1.45839110e-01
3.75929356e-01 1.03442281e-01 3.34191382e-01 7.26516219e-03

```
 3.02870512e-01 3.73373419e-01 1.65178943e-02 6.06051505e-01
 4.26042974e-02 3.34921151e-01 7.84965396e-01 1.35708585e-01
 1.25585413e-02 2.86367238e-01 2.75310665e-01 2.72442877e-01
 4.56278399e-02 1.23302594e-01 2.65966117e-01 2.34146208e-01
 1.32369632e-02 1.51935533e-01 4.03380096e-02 5.57178855e-01
 4.74108100e-01 4.32812572e-01 5.69842756e-01 3.50119770e-01
 4.77863997e-01 3.01848918e-01 4.57466170e-02 2.49867871e-01
 2.33374640e-01 4.47307169e-01 4.10589933e-01 2.00653076e-01
 1.42456263e-01 5.22847287e-03 5.58842540e-01 3.65514070e-01
 9.31568518e-02 2.28247538e-01 5.16135097e-02 3.70492749e-02
 2.71658748e-02 5.67666054e-01 1.88279375e-01 3.75911951e-01
 2.85696685e-01 3.55241150e-02 1.54580295e-01 5.96813679e-01
 2.63294697e-01 4.22281861e-01 2.60984361e-01 3.71472180e-01
 8.90714303e-02 1.69339418e-01 6.47527352e-02 2.12141514e-01
 7.65713215e-01 3.68341178e-01 1.11483045e-01 1.24039099e-01
 1.82616293e-01 2.39016876e-01 6.01025820e-02 6.03920043e-01
 4.82589215e-01 6.23974025e-01 3.26445580e-01 4.97122973e-01
 3.49236578e-02 2.01469213e-01 1.46501541e-01 2.84149855e-01
 9.77053270e-02 1.10063955e-01 1.93680804e-02 1.57768920e-01
 2.33166777e-02 6.67633116e-02 1.22131957e-02 3.76090896e-03
 4.32159930e-01 1.02923743e-01 2.79991329e-01 5.44261113e-02
 2.73895055e-01 5.76362126e-02 1.53888658e-01 5.80461800e-01
 2.90203422e-01 7.38987625e-01 7.08433032e-01 1.56123163e-02
 4.93447781e-01 3.66271995e-02 2.89269209e-01 3.26563925e-01
 4.43824530e-02 1.68435350e-01 2.41399661e-01 5.72726786e-01
 1.40636535e-02 3.71573716e-01 1.66969985e-01 6.22055829e-01
 1.02169402e-01 7.60375023e-01 2.26733580e-01 1.42150121e-02
 6.25262875e-03 2.76215449e-02 2.72625148e-01 4.09550339e-01
 3.66017669e-01 2.53846776e-02 5.25788724e-01 2.25831685e-03
 4.27297264e-01 2.50608381e-03 4.93075132e-01 1.18087552e-01
 1.06275052e-01 5.07073700e-01 3.11444461e-01 1.97149396e-01
 5.24168946e-02 6.02704659e-02 4.91733178e-02 6.72898650e-01
 7.73501918e-02 2.52413768e-02 2.10284859e-01 1.34948194e-01
 3.48658934e-02 3.12642843e-01 3.01034693e-02 5.58365941e-01
 1.40876984e-02 1.40833445e-02 5.80515787e-02 8.95027995e-01
 1.07266746e-01 9.57833044e-03 2.63756383e-02 5.53878963e-01
 6.02728128e-01 3.66285996e-04 3.74555349e-01 1.99137881e-01
 8.76600519e-02 6.17635623e-02 2.93018650e-02 1.49472743e-01
 2.99357008e-02 2.79466242e-01 3.06506038e-01 1.24694213e-01
 1.92340941e-03 8.54413807e-02 1.17961042e-01 8.05716217e-02
 6.10258937e-01 1.09244406e-01 1.28849506e-01 1.31222144e-01
 1.14601031e-01 1.57014076e-02 3.52283657e-01 6.11985065e-02
 2.23112196e-01 7.02888519e-02 7.80055821e-01 3.32020998e-01]

/usr/local/lib/python3.10/dist-packages/numba/cuda/dispatcher.py:536:
NumbaPerformanceWarning: Grid size 4 will likely result in GPU under-
utilization due to low occupancy.
  warn(NumbaPerformanceWarning(msg))
```

Importing Libraries:

- `numpy` is imported to work with arrays in Python.
- `numba.cuda` is imported to utilize CUDA for parallel computing.

Kernel Function Definition (`multiply_elements`):

- This kernel function computes the element-wise product of two input arrays `a` and `b` and stores the result in array `c`.
- It is decorated with `@cuda.jit`, indicating it's a CUDA kernel.
- Inside the kernel:
  - `cuda.grid(1)` is used to retrieve the thread index within a one-dimensional block.
  - Each thread calculates the product of corresponding elements in arrays `a` and `b`.
  - The result is stored in the corresponding position in array `c`.

Main Function (`main()`):

- The main function initializes two input arrays `a` and `b` with random float values.
- Memory for the output array `c` is allocated on the CPU using `np.zeros_like(a)`.
- Memory for arrays `a`, `b`, and `c` is allocated on the GPU using `cuda.to_device()`.
- The grid and block sizes are calculated to determine the number of threads and blocks to launch the kernel.
- The kernel `multiply_elements` is launched on the GPU.
- The result is copied back to the CPU.
- The resulting array `c` containing the element-wise product of `a` and `b` is printed.

Execution:

- When executed, this code performs parallel computation of the element-wise product of two arrays on the GPU using CUDA.
- It efficiently utilizes parallelism provided by CUDA to perform computations faster compared to traditional CPU-based methods.
- The result is then copied back to the CPU for further processing or display.