CS4370

Parallel Programming Many-Core GPUs

Brandon Walker

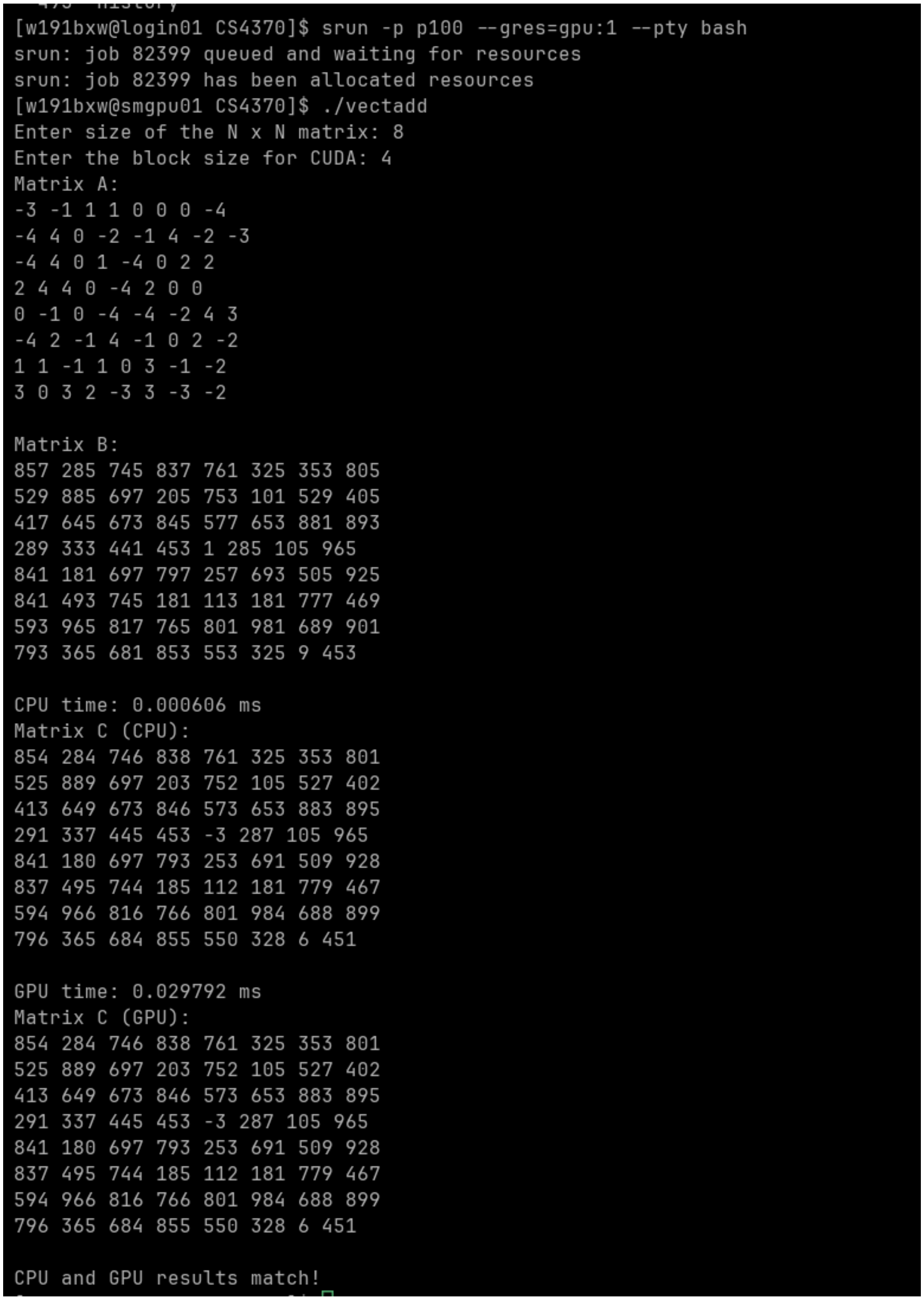
Meilin Liu

23-Sep-2024

Project 1

My matrix addition program has all required functionality, as well as showing timing of cpu vs gpu. It is compiled with the following command. The extra part of the std= etc is necessary because of my timing functionality.

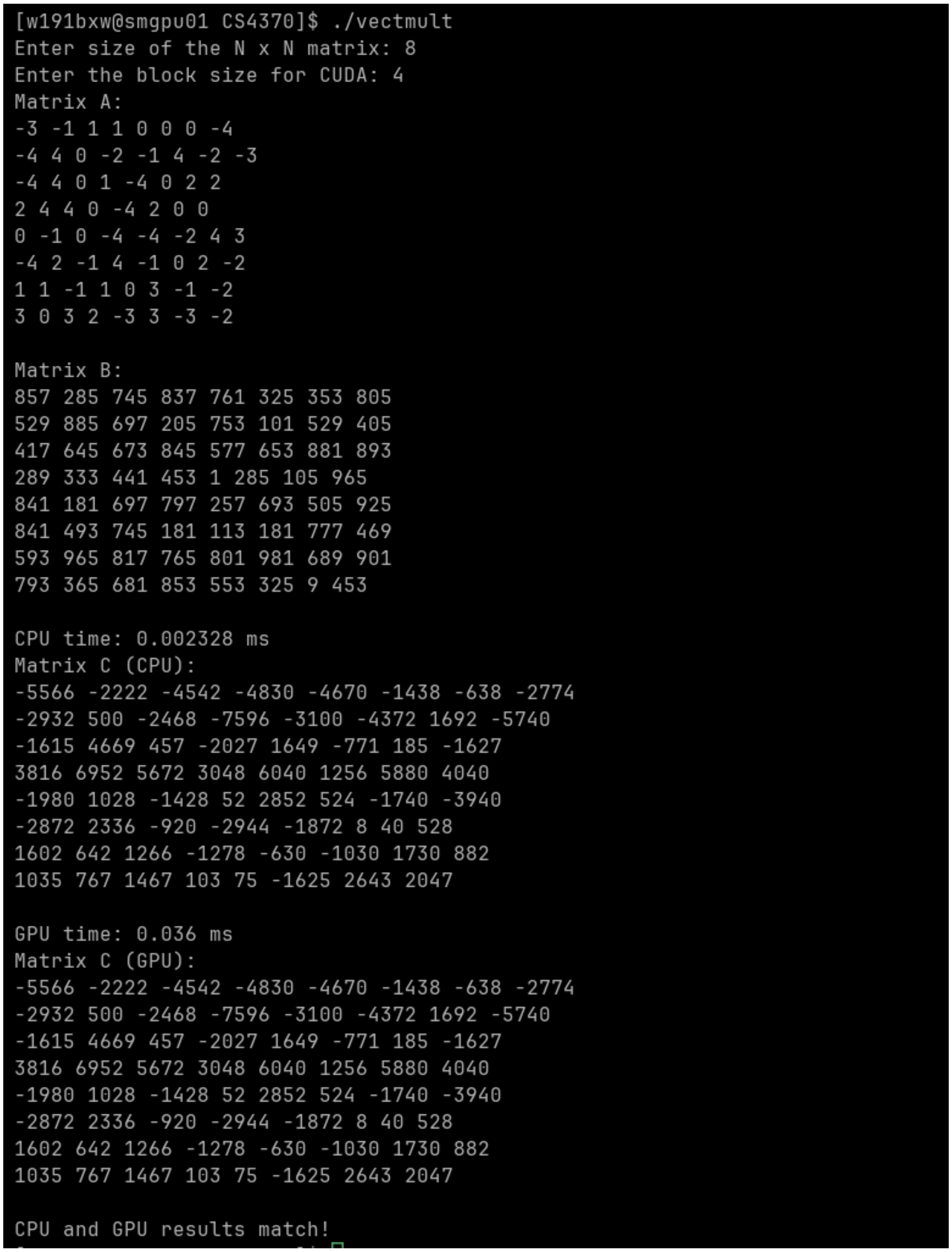
nvcc Walker_vectadd.cu -o vectadd -std=c++11



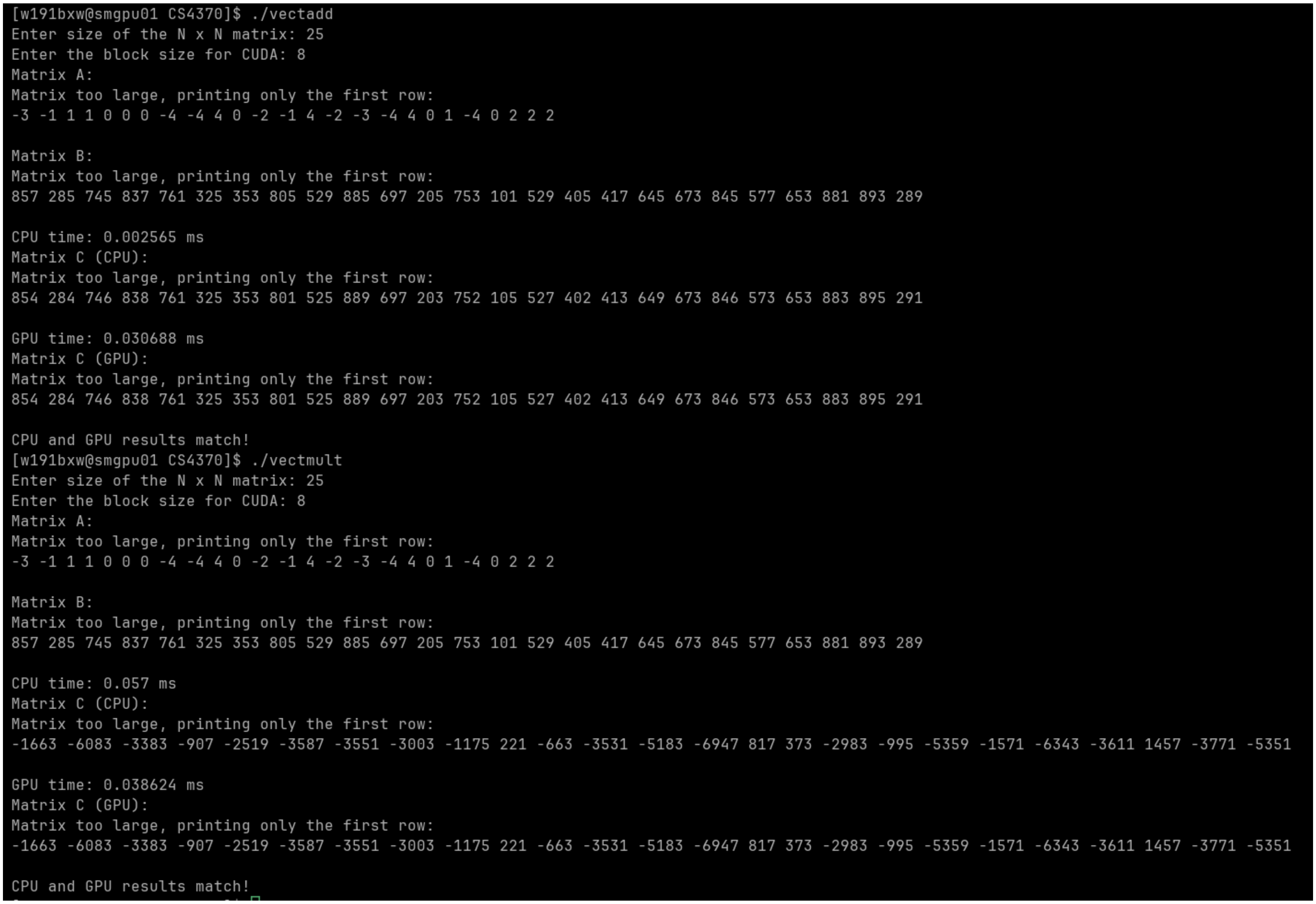
```
  475  history
[w191bxw@login01 CS4370]$ srun -p p100 --gres=gpu:1 --pty bash
srun: job 82399 queued and waiting for resources
srun: job 82399 has been allocated resources
[w191bxw@smgpu01 CS4370]$ ./vectadd
Enter size of the N x N matrix: 8
Enter the block size for CUDA: 4
Matrix A:
-3 -1 1 1 0 0 0 -4
-4 4 0 -2 -1 4 -2 -3
-4 4 0 1 -4 0 2 2
2 4 4 0 -4 2 0 0
0 -1 0 -4 -4 -2 4 3
-4 2 -1 4 -1 0 2 -2
1 1 -1 1 0 3 -1 -2
3 0 3 2 -3 3 -3 -2

Matrix B:
857 285 745 837 761 325 353 805
529 885 697 205 753 101 529 405
417 645 673 845 577 653 881 893
289 333 441 453 1 285 105 965
841 181 697 797 257 693 505 925
841 493 745 181 113 181 777 469
593 965 817 765 801 981 689 901
793 365 681 853 553 325 9 453

CPU time: 0.000606 ms
Matrix C (CPU):
854 284 746 838 761 325 353 801
525 889 697 203 752 105 527 402
413 649 673 846 573 653 883 895
291 337 445 453 -3 287 105 965
841 180 697 793 253 691 509 928
837 495 744 185 112 181 779 467
594 966 816 766 801 984 688 899
796 365 684 855 550 328 6 451

GPU time: 0.029792 ms
Matrix C (GPU):
854 284 746 838 761 325 353 801
525 889 697 203 752 105 527 402
413 649 673 846 573 653 883 895
291 337 445 453 -3 287 105 965
841 180 697 793 253 691 509 928
837 495 744 185 112 181 779 467
594 966 816 766 801 984 688 899
796 365 684 855 550 328 6 451

CPU and GPU results match!
```

My matrix multiplication program has all required functionality, with the added timing, and is compiled with the following command and same logic as the addition command.

```
nvcc Walker_vectmult.cu -o vectmult -std=c++11
```



```
[w191bxw@smgpu01 CS4370]$ ./vectmult
Enter size of the N x N matrix: 8
Enter the block size for CUDA: 4
Matrix A:
-3 -1 1 1 0 0 0 -4
-4 4 0 -2 -1 4 -2 -3
-4 4 0 1 -4 0 2 2
2 4 4 0 -4 2 0 0
0 -1 0 -4 -4 -2 4 3
-4 2 -1 4 -1 0 2 -2
1 1 -1 1 0 3 -1 -2
3 0 3 2 -3 3 -3 -2

Matrix B:
857 285 745 837 761 325 353 805
529 885 697 205 753 101 529 405
417 645 673 845 577 653 881 893
289 333 441 453 1 285 105 965
841 181 697 797 257 693 505 925
841 493 745 181 113 181 777 469
593 965 817 765 801 981 689 901
793 365 681 853 553 325 9 453

CPU time: 0.002328 ms
Matrix C (CPU):
-5566 -2222 -4542 -4830 -4670 -1438 -638 -2774
-2932 500 -2468 -7596 -3100 -4372 1692 -5740
-1615 4669 457 -2027 1649 -771 185 -1627
3816 6952 5672 3048 6040 1256 5880 4040
-1980 1028 -1428 52 2852 524 -1740 -3940
-2872 2336 -920 -2944 -1872 8 40 528
1602 642 1266 -1278 -630 -1030 1730 882
1035 767 1467 103 75 -1625 2643 2047

GPU time: 0.036 ms
Matrix C (GPU):
-5566 -2222 -4542 -4830 -4670 -1438 -638 -2774
-2932 500 -2468 -7596 -3100 -4372 1692 -5740
-1615 4669 457 -2027 1649 -771 185 -1627
3816 6952 5672 3048 6040 1256 5880 4040
-1980 1028 -1428 52 2852 524 -1740 -3940
-2872 2336 -920 -2944 -1872 8 40 528
1602 642 1266 -1278 -630 -1030 1730 882
1035 767 1467 103 75 -1625 2643 2047

CPU and GPU results match!
```



```
[w191bxw@smgpu01 CS4370]$ ./vectadd
Enter size of the N x N matrix: 25
Enter the block size for CUDA: 8
Matrix A:
Matrix too large, printing only the first row:
-3 -1 1 1 0 0 0 -4 -4 4 0 -2 -1 4 -2 -3 -4 4 0 1 -4 0 2 2 2

Matrix B:
Matrix too large, printing only the first row:
857 285 745 837 761 325 353 805 529 885 697 205 753 101 529 405 417 645 673 845 577 653 881 893 289

CPU time: 0.002565 ms
Matrix C (CPU):
Matrix too large, printing only the first row:
854 284 746 838 761 325 353 801 525 889 697 203 752 105 527 402 413 649 673 846 573 653 883 895 291

GPU time: 0.030688 ms
Matrix C (GPU):
Matrix too large, printing only the first row:
854 284 746 838 761 325 353 801 525 889 697 203 752 105 527 402 413 649 673 846 573 653 883 895 291

CPU and GPU results match!
[w191bxw@smgpu01 CS4370]$ ./vectmult
Enter size of the N x N matrix: 25
Enter the block size for CUDA: 8
Matrix A:
Matrix too large, printing only the first row:
-3 -1 1 1 0 0 0 -4 -4 4 0 -2 -1 4 -2 -3 -4 4 0 1 -4 0 2 2 2

Matrix B:
Matrix too large, printing only the first row:
857 285 745 837 761 325 353 805 529 885 697 205 753 101 529 405 417 645 673 845 577 653 881 893 289

CPU time: 0.057 ms
Matrix C (CPU):
Matrix too large, printing only the first row:
-1663 -6083 -3383 -907 -2519 -3587 -3551 -3003 -1175 221 -663 -3531 -5183 -6947 817 373 -2983 -995 -5359 -1571 -6343 -3611 1457 -3771 -5351

GPU time: 0.038624 ms
Matrix C (GPU):
Matrix too large, printing only the first row:
-1663 -6083 -3383 -907 -2519 -3587 -3551 -3003 -1175 221 -663 -3531 -5183 -6947 817 373 -2983 -995 -5359 -1571 -6343 -3611 1457 -3771 -5351

CPU and GPU results match!
```
]]>