Yakshita B Rakholiya
yr92282n@pace.edu
Student ID: U01875270
Course: CS-610-22756
Project-7

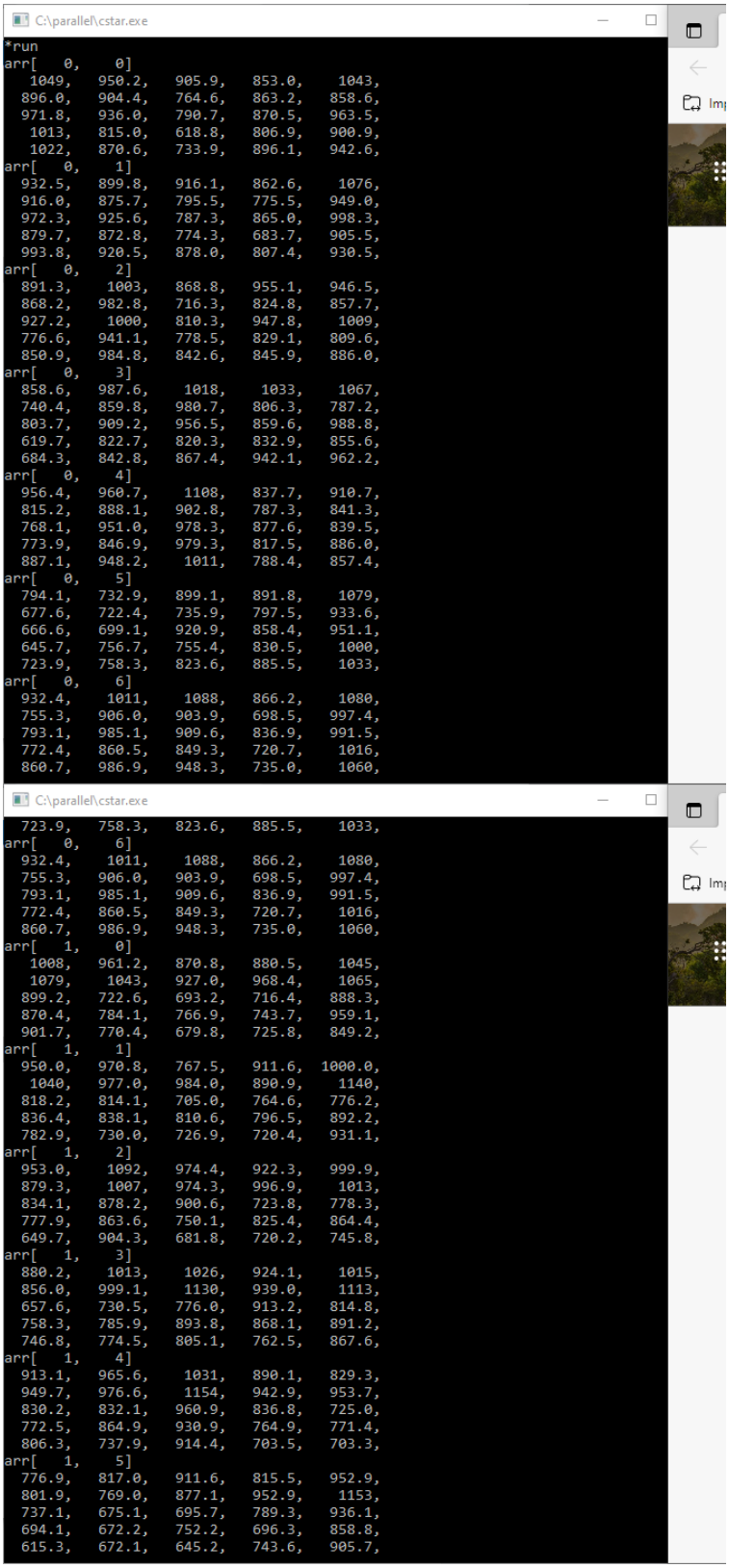
Developing an efficient parallel matrix multiplication program on a multicomputer with a torus interconnection network.

C:\parallel\cstar.exe

```
*open matrix.c


Program Successfully Compiled

To View a Complete Program Listing, See File LISTFILE.TXT

*view
   1 /*
   2 Pace University CS610
   3 Yakshita Rakholiya
   4 Project-6 @Dr.Lixin Tao @Kai Wang
   5 */
   6
   7 #include <stdlib.h>
   8
   9
  10 #define p_arr_size 7
  11 #define p_arr 5
  12
  13 typedef float arr[p_arr+1][p_arr+1];
  14 arr a[p_arr_size][p_arr_size], b[p_arr_size][p_arr_size], c[p_arr_size][p_arr
_size];
  15 arr stream One[p_arr_size][p_arr_size], stream Two[p_arr_size][p_arr_size];
  16 int x,y;
  17
  18 void my_func(int i, int j,
  19        value arr var_A, value arr var_B,
  20       arr var_C) {
  21   int x,y,k,iter,u,l;
  22   arr myC = {0};
  23   if (i > 0) u = i-1; else u = p_arr_size-1;
  24   if (j > 0) l = j-1; else l = p_arr_size-1;
  25   for (iter = 1; iter <= p_arr_size; iter++) {
  26     send(One[i][l], var_A);
  27     send(Two[u][j], var_B);
  28
  29     for (x = 1; x <= p_arr; x++)
  30       for (y = 1; y <= p_arr; y++)
  31             for (k = 1; k <= p_arr; k++)
  32               myC[x][y] = myC[x][y] + var_A[x][k]*var_B[k][y];
  33     recv(One[i][j], var_A);
  34     recv(Two[i][j], var_B);
```

C:\parallel\cstar.exe

```
  35   }
  36   var_C = myC;
  37 }
  38
  39 main( ) {
  40
  41   int k, l;
  42   for (x = 0; x < p_arr_size; x++)
  43     for (y = 0; y < p_arr_size; y++)
  44       for (k = 1; k <= p_arr; k++)
  45         for (l = 1; l <= p_arr; l++) {
  46               a[x][y][k][l] = (rand() % 100)/10.0;
  47               b[x][y][k][l] = (rand() % 100)/10.0;
  48       }
  49
  50
  51   for (x = 0; x < p_arr_size; x++)
  52     for (y = 0; y < p_arr_size; y++)
  53       fork (@x*p_arr_size+y)
  54          my_func(x, y, a[x][(y+x)%p_arr_size], b[(x+y)%p_arr_size][y], c[x
);
  55
  56   for (x = 0; x < p_arr_size; x++)
  57     for (y = 0; y < p_arr_size; y++)
  58       join;
  59
  60   cout.precision(4);
  61   for (x = 0; x < p_arr_size; x++) {
  62     for (y = 0; y < p_arr_size; y++) {
  63       cout << "arr["<< x << ", " << y <<"]" << endl;
  64       for (k = 1; k <= p_arr; k++) {
  65             for (l = 1; l <= p_arr; l++)
  66               cout << c[x][y][k][l] << ", ";
  67         cout << endl;
  68       }
  69     }
  70   }
  71 }
  72
  73
  74

*run
```


C:\parallel\cstar.exe
*run
arr[0, 0]
1049, 950.2, 905.9, 853.0, 1043,
896.0, 904.4, 764.6, 863.2, 858.6,
971.8, 936.0, 790.7, 870.5, 963.5,
1013, 815.0, 618.8, 806.9, 900.9,
1022, 870.6, 733.9, 896.1, 942.6,
arr[0, 1]
932.5, 899.8, 916.1, 862.6, 1076,
916.0, 875.7, 795.5, 775.5, 949.0,
972.3, 925.6, 787.3, 865.0, 998.3,
879.7, 872.8, 774.3, 683.7, 905.5,
993.8, 920.5, 878.0, 807.4, 930.5,
arr[0, 2]
891.3, 1003, 868.8, 955.1, 946.5,
868.2, 982.8, 716.3, 824.8, 857.7,
927.2, 1000, 810.3, 947.8, 1009,
776.6, 941.1, 778.5, 829.1, 809.6,
850.9, 984.8, 842.6, 845.9, 886.0,
arr[0, 3]
858.6, 987.6, 1018, 1033, 1067,
740.4, 859.8, 980.7, 806.3, 787.2,
803.7, 909.2, 956.5, 859.6, 988.8,
619.7, 822.7, 820.3, 832.9, 855.6,
684.3, 842.8, 867.4, 942.1, 962.2,
arr[0, 4]
956.4, 960.7, 1108, 837.7, 910.7,
815.2, 888.1, 902.8, 787.3, 841.3,
768.1, 951.0, 978.3, 877.6, 839.5,
773.9, 846.9, 979.3, 817.5, 886.0,
887.1, 948.2, 1011, 788.4, 857.4,
arr[0, 5]
794.1, 732.9, 899.1, 891.8, 1079,
677.6, 722.4, 735.9, 797.5, 933.6,
666.6, 699.1, 920.9, 858.4, 951.1,
645.7, 756.7, 755.4, 830.5, 1000,
723.9, 758.3, 823.6, 885.5, 1033,
arr[0, 6]
932.4, 1011, 1088, 866.2, 1080,
755.3, 906.0, 903.9, 698.5, 997.4,
793.1, 985.1, 909.6, 836.9, 991.5,
772.4, 860.5, 849.3, 720.7, 1016,
860.7, 986.9, 948.3, 735.0, 1060,
C:\parallel\cstar.exe
723.9, 758.3, 823.6, 885.5, 1033,
arr[0, 6]
932.4, 1011, 1088, 866.2, 1080,
755.3, 906.0, 903.9, 698.5, 997.4,
793.1, 985.1, 909.6, 836.9, 991.5,
772.4, 860.5, 849.3, 720.7, 1016,
860.7, 986.9, 948.3, 735.0, 1060,
arr[1, 0]
1008, 961.2, 870.8, 880.5, 1045,
1079, 1043, 927.0, 968.4, 1065,
899.2, 722.6, 693.2, 716.4, 888.3,
870.4, 784.1, 766.9, 743.7, 959.1,
901.7, 770.4, 679.8, 725.8, 849.2,
arr[1, 1]
950.0, 970.8, 767.5, 911.6, 1000.0,
1040, 977.0, 984.0, 890.9, 1140,
818.2, 814.1, 705.0, 764.6, 776.2,
836.4, 838.1, 810.6, 796.5, 892.2,
782.9, 730.0, 726.9, 720.4, 931.1,
arr[1, 2]
953.0, 1092, 974.4, 922.3, 999.9,
879.3, 1007, 974.3, 996.9, 1013,
834.1, 878.2, 900.6, 723.8, 778.3,
777.9, 863.6, 750.1, 825.4, 864.4,
649.7, 904.3, 681.8, 720.2, 745.8,
arr[1, 3]
880.2, 1013, 1026, 924.1, 1015,
856.0, 999.1, 1130, 939.0, 1113,
657.6, 730.5, 776.0, 913.2, 814.8,
758.3, 785.9, 893.8, 868.1, 891.2,
746.8, 774.5, 805.1, 762.5, 867.6,
arr[1, 4]
913.1, 965.6, 1031, 890.1, 829.3,
949.7, 976.6, 1154, 942.9, 953.7,
830.2, 832.1, 960.9, 836.8, 725.0,
772.5, 864.9, 930.9, 764.9, 771.4,
806.3, 737.9, 914.4, 703.5, 703.3,
arr[1, 5]
776.9, 817.0, 911.6, 815.5, 952.9,
801.9, 769.0, 877.1, 952.9, 1153,
737.1, 675.1, 695.7, 789.3, 936.1,
694.1, 672.2, 752.2, 696.3, 858.8,
615.3, 672.1, 645.2, 743.6, 905.7,