Nithishkumar.K

EX.1 Ex.No:-6

Implementing matrix multiplication using open-MP

Nithishkumar.K.
18CO66

Aim

To implement matrix multiplication using open-MP framework parallely.

Description

open MP is a library for parallel computing. It support shared memory so all threads share the memory. It is cross platform and is available in languages like C, C++, fortran.

Steps to implement

- write a program matrix multiply. modify it for OMP by adding the # program segement (which is used for parallelity

- Code that is present inside the program segement will be run in multiple processor.

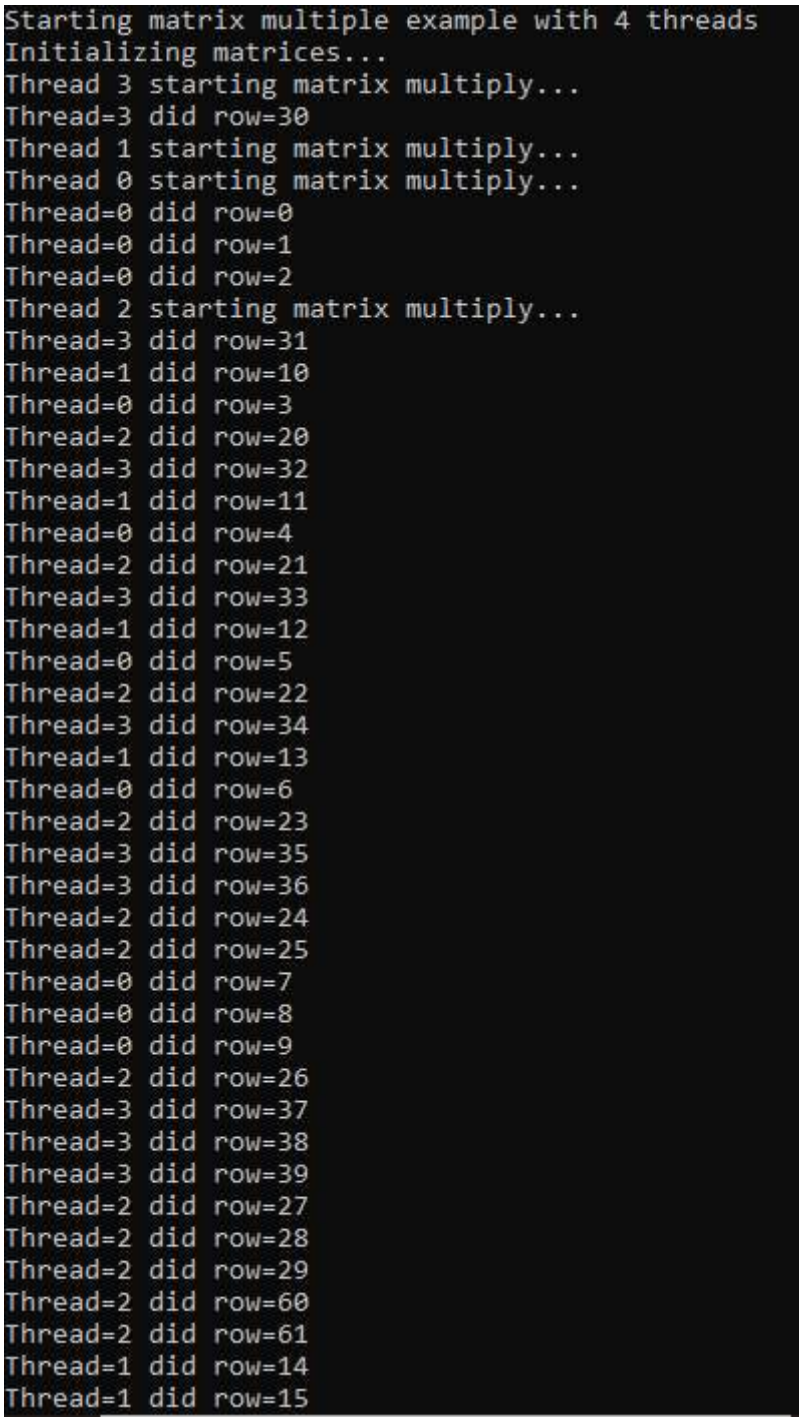
- The individual multiplications can be parallelised and hence the performance can be improved.

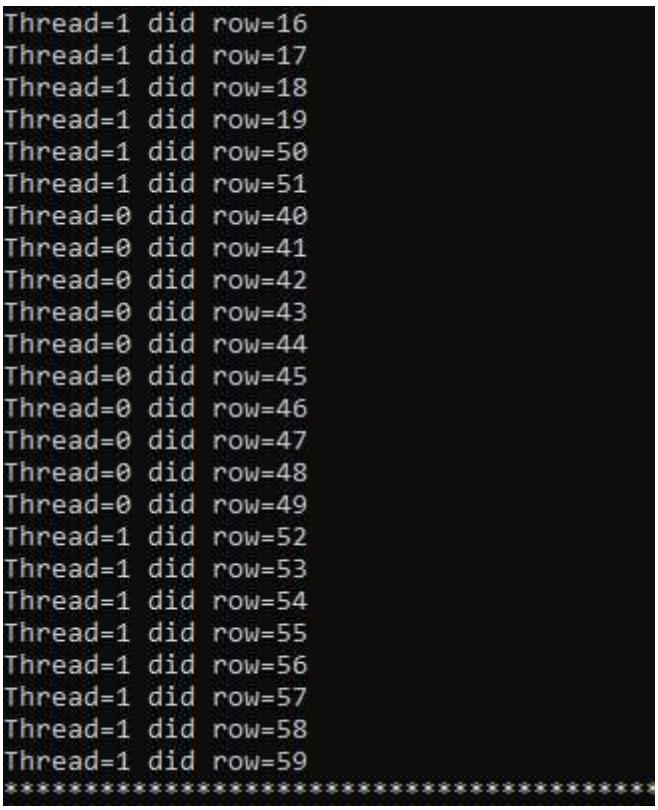
OPENMP CODE:

```
#include <omp.h>
#include <stdio.h>
#include <stdlib.h>
#define NRA 62
#define NCA 15
#define NCB 7
int main (int argc, char *argv[])
{
int tid, nthreads, i, j, k, chunk;
double a[NRA][NCA],
 b[NCA][NCB],
 c[NRA][NCB];
chunk = 10;

#pragma omp parallel shared(a,b,c,nthreads,chunk) private(tid,i,j,k)
 {
 tid = omp_get_thread_num();
 if (tid == 0)
 {
 nthreads = omp_get_num_threads();
 printf("Starting matrix multiple example with %d threads\n",nthreads);
 printf("Initializing matrices...\n");
 }

 #pragma omp for schedule (static, chunk)
 for (i=0; i<NRA; i++)
 for (j=0; j<NCA; j++)
 a[i][j]= i+j;
 #pragma omp for schedule (static, chunk)
 for (i=0; i<NCA; i++)
```

```
 for (j=0; j<NCB; j++)
 b[i][j]= i*j;
 #pragma omp for schedule (static, chunk)
 for (i=0; i<NRA; i++)
 for (j=0; j<NCB; j++)
 c[i][j]= 0;


 printf("Thread %d starting matrix multiply...\n",tid);
 #pragma omp for schedule (static, chunk)
 for (i=0; i<NRA; i++)
 {
 printf("Thread=%d did row=%d\n",tid,i);
 for(j=0; j<NCB; j++)
 for (k=0; k<NCA; k++)
 c[i][j] += a[i][k] * b[k][j];
 }
 }

printf("******************************************************\n");
printf("Result Matrix:\n");
for (i=0; i<NRA; i++)
 {
 for (j=0; j<NCB; j++)
 printf("%6.2f ", c[i][j]);
 printf("\n");
 }
printf("******************************************************\n");
printf ("Done.\n");
}
```

OUTPUT:


Starting matrix multiple example with 4 threads
Initializing matrices...
Thread 3 starting matrix multiply...
Thread=3 did row=30
Thread 1 starting matrix multiply...
Thread 0 starting matrix multiply...
Thread=0 did row=0
Thread=0 did row=1
Thread=0 did row=2
Thread 2 starting matrix multiply...
Thread=3 did row=31
Thread=1 did row=10
Thread=0 did row=3
Thread=2 did row=20
Thread=3 did row=32
Thread=1 did row=11
Thread=0 did row=4
Thread=2 did row=21
Thread=3 did row=33
Thread=1 did row=12
Thread=0 did row=5
Thread=2 did row=22
Thread=3 did row=34
Thread=1 did row=13
Thread=0 did row=6
Thread=2 did row=23
Thread=3 did row=35
Thread=3 did row=36
Thread=2 did row=24
Thread=2 did row=25
Thread=0 did row=7
Thread=0 did row=8
Thread=0 did row=9
Thread=2 did row=26
Thread=3 did row=37
Thread=3 did row=38
Thread=3 did row=39
Thread=2 did row=27
Thread=2 did row=28
Thread=2 did row=29
Thread=2 did row=60
Thread=2 did row=61
Thread=1 did row=14
Thread=1 did row=15


Thread=1 did row=16
Thread=1 did row=17
Thread=1 did row=18
Thread=1 did row=19
Thread=1 did row=50
Thread=1 did row=51
Thread=0 did row=40
Thread=0 did row=41
Thread=0 did row=42
Thread=0 did row=43
Thread=0 did row=44
Thread=0 did row=45
Thread=0 did row=46
Thread=0 did row=47
Thread=0 did row=48
Thread=0 did row=49
Thread=1 did row=52
Thread=1 did row=53
Thread=1 did row=54
Thread=1 did row=55
Thread=1 did row=56
Thread=1 did row=57
Thread=1 did row=58
Thread=1 did row=59
****

```
Result Matrix:
  0.00 1015.00 2030.00 3045.00 4060.00 5075.00 6090.00
  0.00 1120.00 2240.00 3360.00 4480.00 5600.00 6720.00
  0.00 1225.00 2450.00 3675.00 4900.00 6125.00 7350.00
  0.00 1330.00 2660.00 3990.00 5320.00 6650.00 7980.00
  0.00 1435.00 2870.00 4305.00 5740.00 7175.00 8610.00
  0.00 1540.00 3080.00 4620.00 6160.00 7700.00 9240.00
  0.00 1645.00 3290.00 4935.00 6580.00 8225.00 9870.00
  0.00 1750.00 3500.00 5250.00 7000.00 8750.00 10500.00
  0.00 1855.00 3710.00 5565.00 7420.00 9275.00 11130.00
  0.00 1960.00 3920.00 5880.00 7840.00 9800.00 11760.00
  0.00 2065.00 4130.00 6195.00 8260.00 10325.00 12390.00
  0.00 2170.00 4340.00 6510.00 8680.00 10850.00 13020.00
  0.00 2275.00 4550.00 6825.00 9100.00 11375.00 13650.00
  0.00 2380.00 4760.00 7140.00 9520.00 11900.00 14280.00
  0.00 2485.00 4970.00 7455.00 9940.00 12425.00 14910.00
  0.00 2590.00 5180.00 7770.00 10360.00 12950.00 15540.00
  0.00 2695.00 5390.00 8085.00 10780.00 13475.00 16170.00
  0.00 2800.00 5600.00 8400.00 11200.00 14000.00 16800.00
  0.00 2905.00 5810.00 8715.00 11620.00 14525.00 17430.00
  0.00 3010.00 6020.00 9030.00 12040.00 15050.00 18060.00
  0.00 3115.00 6230.00 9345.00 12460.00 15575.00 18690.00
  0.00 3220.00 6440.00 9660.00 12880.00 16100.00 19320.00
  0.00 3325.00 6650.00 9975.00 13300.00 16625.00 19950.00
  0.00 3430.00 6860.00 10290.00 13720.00 17150.00 20580.00
  0.00 3535.00 7070.00 10605.00 14140.00 17675.00 21210.00
  0.00 3640.00 7280.00 10920.00 14560.00 18200.00 21840.00
  0.00 3745.00 7490.00 11235.00 14980.00 18725.00 22470.00
  0.00 3850.00 7700.00 11550.00 15400.00 19250.00 23100.00
  0.00 3955.00 7910.00 11865.00 15820.00 19775.00 23730.00
  0.00 4060.00 8120.00 12180.00 16240.00 20300.00 24360.00
  0.00 4165.00 8330.00 12495.00 16660.00 20825.00 24990.00
  0.00 4270.00 8540.00 12810.00 17080.00 21350.00 25620.00
  0.00 4375.00 8750.00 13125.00 17500.00 21875.00 26250.00
  0.00 4480.00 8960.00 13440.00 17920.00 22400.00 26880.00
  0.00 4585.00 9170.00 13755.00 18340.00 22925.00 27510.00
  0.00 4690.00 9380.00 14070.00 18760.00 23450.00 28140.00
  0.00 4795.00 9590.00 14385.00 19180.00 23975.00 28770.00
  0.00 4900.00 9800.00 14700.00 19600.00 24500.00 29400.00
  0.00 5005.00 10010.00 15015.00 20020.00 25025.00 30030.00
  0.00 5110.00 10220.00 15330.00 20440.00 25550.00 30660.00
  0.00 5215.00 10430.00 15645.00 20860.00 26075.00 31290.00
  0.00 5320.00 10640.00 15960.00 21280.00 26600.00 31920.00
  0.00 5425.00 10850.00 16275.00 21700.00 27125.00 32550.00
  0.00 6160.00 12320.00 18480.00 24640.00 30800.00 36960.00
  0.00 6265.00 12530.00 18795.00 25060.00 31325.00 37590.00
  0.00 6370.00 12740.00 19110.00 25480.00 31850.00 38220.00
  0.00 6475.00 12950.00 19425.00 25900.00 32375.00 38850.00
  0.00 6580.00 13160.00 19740.00 26320.00 32900.00 39480.00
  0.00 6685.00 13370.00 20055.00 26740.00 33425.00 40110.00
  0.00 6790.00 13580.00 20370.00 27160.00 33950.00 40740.00
  0.00 6895.00 13790.00 20685.00 27580.00 34475.00 41370.00
  0.00 7000.00 14000.00 21000.00 28000.00 35000.00 42000.00
  0.00 7105.00 14210.00 21315.00 28420.00 35525.00 42630.00
  0.00 7210.00 14420.00 21630.00 28840.00 36050.00 43260.00
  0.00 7315.00 14630.00 21945.00 29260.00 36575.00 43890.00
  0.00 7420.00 14840.00 22260.00 29680.00 37100.00 44520.00
************************************************************
Done.
```

Result:

Thus, the parallel program for matrix multiplication using open mp constructs is developed.