Rohit Kulkarni 41346

M T W T F S S
Page No.:
Date:
YOUVA

HPC ASSIGNMENT 2

TITLE:
Vector & Matrix Operations: Design parallel algorithm to-
1] Add two large vectors.
2] Multiple vector & matrix.
3] Multiple two NXN array using N^2 processors.

OBJECTWES:
1] Learn parapllel programming using OpenMp.

OUTCOMES:
1] Design parallel algorithm.
2] Use OpenMp for task parallelism.

THEORY:
1) OpenMP is an API that can be used with FORTRAN, C & C++ for programming shared address space machines.
2) OpenMP directives provide support for concurrency, synchronization & data handling while obviating the need for explicitly setting up mutexes, condition variables, data scope & initialization.
3) OpenMP directives in C & C++ are based on the #pragma compiler directives.
4) The directive itself consists of a directives name followed by the clauses list.

#pragma omp directive [clause list]

* Matrix Vector Product

i) To define multiplication between a matrix & vector we need to view the vector as column matrix.

ii) We define the matrix-vector product only for the case when the number of columns in vector equals the no. of rows in matrix.

$$\underset{m \times n}{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n} \\ \vdots \\ a_{m1}x_1 + \cdots a_{mn}x_n \end{bmatrix}$$

* ALGORITHM (Matrix-Vector Multiplication)

1] Read the size of matrix MXN & read the numbers randomly.
2] Read ~~ra~~ elements randomly for column vector of size N.
3] Read start time.
4] Using #pragma omp parallel for parallize multiplication row by row.
5] Read end time.
6] Display execution time as end time - start time.

* ALGORITHM (Matrix-Matrix Multiplication)

1] Read the size of matrix NXN & numbers randomly
2] Read the start time.
3] Using #pragma omp parallel for parallelize multiplication row by row.
4] Read end time.
5] Display execution time as end time - start time

M T W T F S S
Page No.:
Date:
YOUVA

* ALGORITHM (Vector addition)

1] Read the size of vector N & read numbers randomly.
2] Read the start time.
3] Using #pragma omp parallel for parallelize single for loop & perform vector addition.
4] Read the end time.
5] Display the execution time as end time - start time.

CONCLUSION:

OpenMP techniques were applied for the given task & data parallelism was studied successfully.