CS 301
High-Performance Computing

Lab 5: Problem A-1

Harsh Makwana (202001264)
Vivek Godhasara (202001451)

March 23, 2023

Contents

1 Introduction **3**
1.1 Brief description of the problem. 3
1.2 The complexity of the algorithm (serial). 3

2 Hardware Details **3**
2.1 Hardware Details of LAB207 Computer 3
2.2 Hardware Details of Cluster 3

3 PART 1: LAB207 Computer **4**
3.1 Graph 4

4 PART 2: Cluster **4**
4.1 Graph 4

1 Introduction

1.1 Brief description of the problem.

Problem A-1 -> Conventional matrix multiplication.

In this problem, our task is to multiply two matrices with size N using the conventional matrix multiplication method. In the conventional method, we use three loops to multiply the rows with the columns. But In this task, we will be writing a parallel algorithm using pragma omp.

Now our goal is to write an optimal parallel algorithm for calculating the upper problem so that we can get optimal use of our processor.

1.2 The complexity of the algorithm (serial).

As you can see in the uploaded code, We are doing matrix multiplication using the conventional method i.e. using three loops.

So The time complexity of the algorithm is:

$$O(N^3)$$

and The space complexity of the algorithm is:

$$O(N^2)$$

2 Hardware Details

2.1 Hardware Details of LAB207 Computer

- CPU - 4
- Socket - 1
- Cores per Socket - 4
- Size of L1 cache - 64KB
- Size of L2 cache - 256KB
- Size of L3 cache - 6MB

2.2 Hardware Details of Cluster

- CPU - 16
- Socket - 2
- Cores per Socket - 8
- Size of L1 cache - 64KB

- Size of L2 cache - 256KB
- Size of L3 cache - 20MB

3 PART 1: LAB207 Computer

3.1 Graph

Below we have depicted the Mean execution time vs problem size for Lab207 Computer.

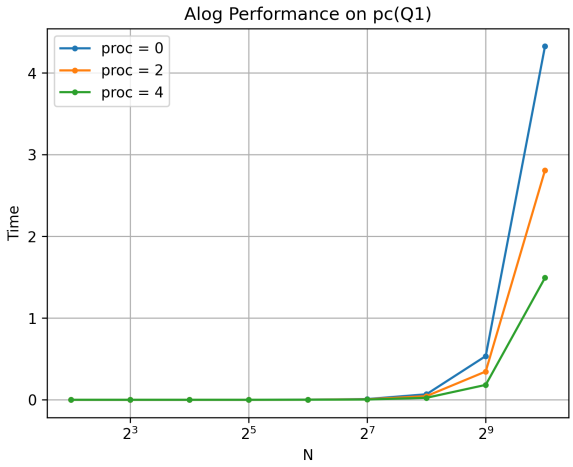

Figure 1: Algorithm time vs problem size - PC

4 PART 2: Cluster

4.1 Graph

Below we have depicted the Mean execution time vs problem size for Cluster.

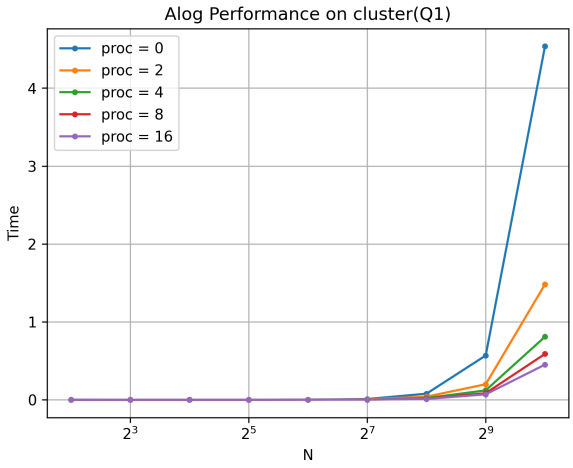

Figure 2: Algorithm time vs problem size - Cluster