CS 301
High-Performance Computing

Lab 4: Problem C2

Vivek Godhasara (202001451)
Harsh Makwana (202001264)

March 15, 2023

Contents

1 Introduction

1.1 Brief description of the problem.

Problem C-2 -> Sorting - Merge Sort.

In this problem, our task is to Sort a given array with size N. According to the instruction in sorting, we will be using the Merge sort algorithm as a sorting algorithm.

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

1.2 The complexity of the algorithm (serial).

As you can see in the uploaded code, In this algorithm we are using a while loop for sorting the array, which is dividing the array into two halves and then using O(N) function so it has total N*log(N) time complexity, where N is the array size.

So The time complexity of the algorithm is: O(N*log(N))
and The space complexity of the algorithm is: O(N)

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.

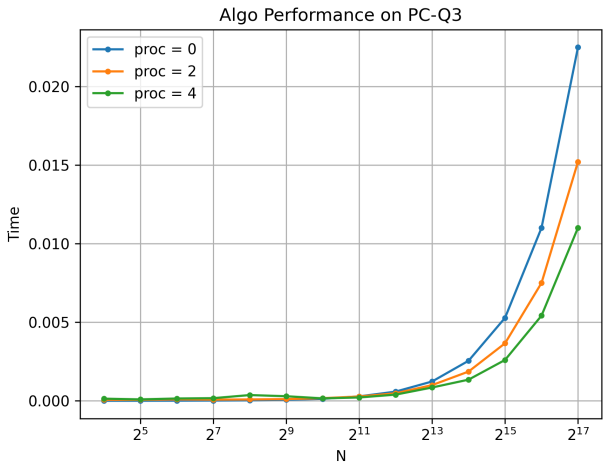

Figure 1: Algorithm time vs problem size - PC

4 PART 2: Cluster

4.1 Graph

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

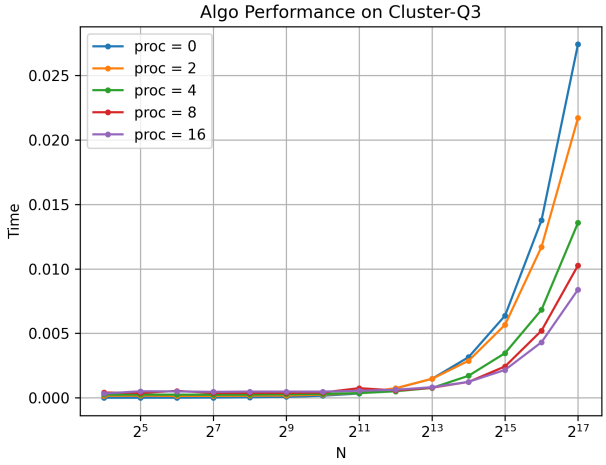

Figure 2: Algorithm time vs problem size - Cluster