សាកលវិទ្យាល័យភូមិន្ទភ្នំពេញ

ROYAL UNIVERSITY OF PHNOM PENH

Multi-threaded and Single-threaded Program

Lecturer: Kor Sokchea

Submitted by: Tang Lymeng

January 2023

TABLE OF CONTENTS

I. Introduction

In this project I will show the comparison between multi-threaded and single-threaded for sorting number by using Java threading for multi-threaded and bubble sorting algorithm for both multi-threaded and single-threaded. I will compare the performance of this sorting algorithm with respect to time and number of input with multi-threaded and single-threaded.

II. Objective

This project will mainly focus on the difference between multi-threaded and single-threaded for sorting number. The performance comparison will be observed by the implementation of bubble sorted algorithm.

III. Implementation

3.1 Multi-threaded sorting program

To implement the multi-threaded sorting program, we will first create a global array of integers and divide it into two sub lists of equal size. We will then create two separate threads, one for each sub list, and pass the starting index for each thread as a parameter. These sorting threads will then use the bubble sorting algorithm to sort the sub lists. Once the both threads done their works, we will create a third threaded and named it merging threaded, which will merge the two sub lists into single list.

3.2 Single-threaded sorting program

For single-threaded sorting program, we will simply implement the bubble sorting algorithm in the main function without creating any threaded.

- Bubble sort algorithm

  Sorting is a technique that arrange names in alphabetical order, put elements of a list into an order, etc.

  - For example of unordered and ordered data

The Unordered (by Height) Baseball Team → *The Ordered (by Height) Baseball Team*

  - How Bubble sort work?

    1. You start at the left end of the line and compare the two kids in positions 0 and 1.

2. If the kid on the left (in position 0) is taller, you swap them. If the kid on the right (in position 1) is taller, you don’t do anything.
3. Then you move over one position and compare the kids in positions 1 and 2.
Again, if the kid on the left is taller, you swap them.

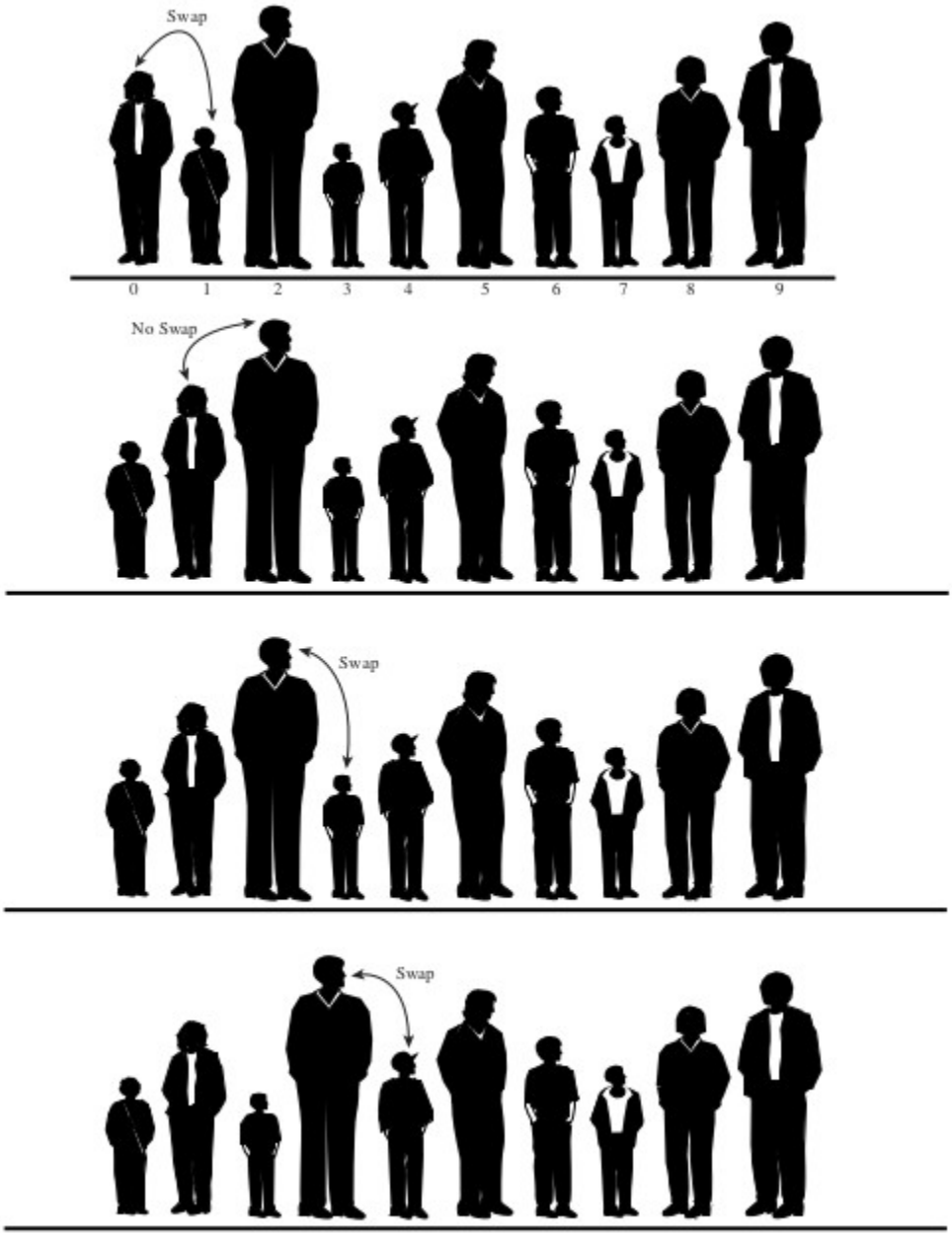


Figure 1: beginning of first pass

Here are the rules you’re following:
1. Compare two players.
2. If the one on the left is taller, swap them.
3. Move one position right.

Figure 2: end of first pass

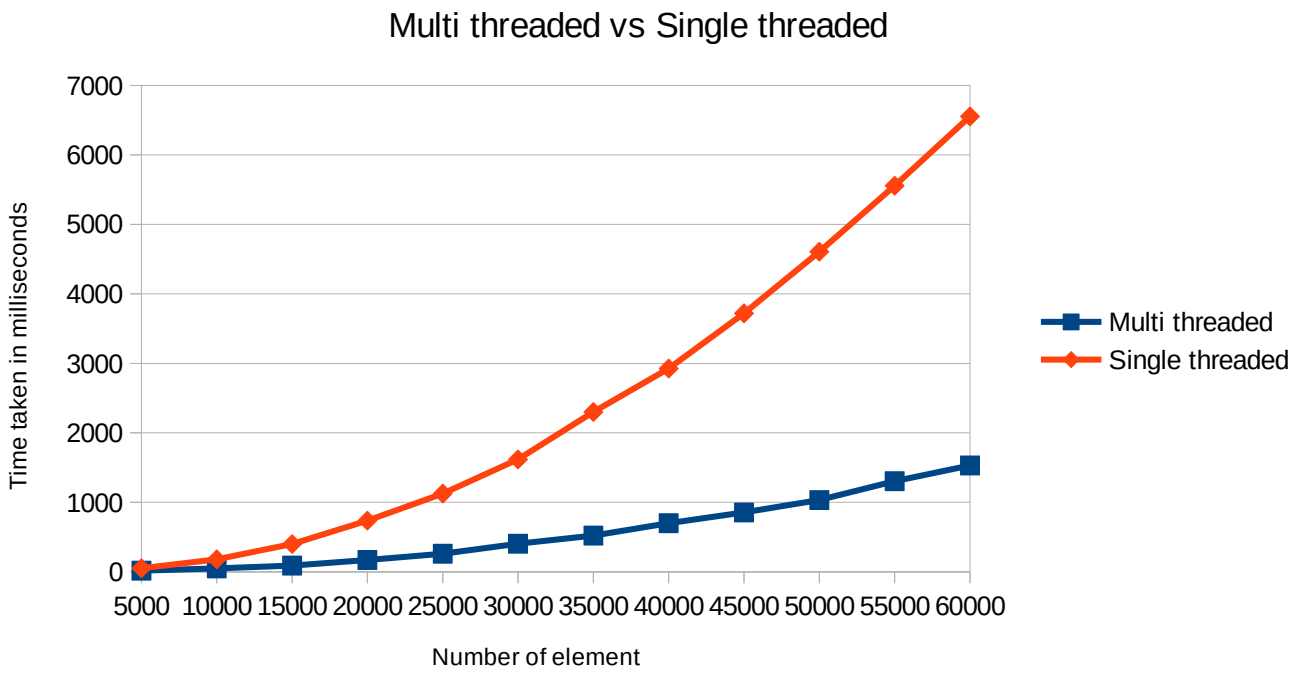
4. When you reach the first already-sorted player, start over at the left end of the line.

Continue this process until all the players are in order.

IV. Result

I have testing both of the program multi-threaded and single threaded with random number between 1-1000.

No. of element(random number 1-1000)	Multi threaded	Single threaded
5000	17	52
10000	49	180
15000	90	400
20000	170	737
25000	261	1126
30000	404	1618
35000	522	2300
40000	700	2926
45000	855	3721
50000	1033	4607
55000	1304	5556
60000	1531	6555



V. Conclusion

After we seen the plotting chart in the result section we can drawn the conclusion that multi-threaded is working 4 time faster than single-threaded. The result is that on a multiprocessor system, multi-threading can concurrently run on multiple CPUs. In the multi-threaded sorting program we split the array into two sub lists so on multi-threaded we can maximize the usage of CPUs.