Performance Analysis using thread of Programming Language C++,Java and Python

Diksha Pande

December 2020

1 Abstract

Nowadays, Multi-threading is becoming increasingly important, both as a program structuring mechanism and to support efficient parallel computations.In this paper we tested the multi-threading performance of C++,Java and Python, these three are the most popular languages used for Software Development in industry.After the performance analysis of these 3 we came to know that the time taken to compute a multi-threading program for C++ is less than time taken by Java and Python.The obtained results provide insight into the current state of optimization of each of the three languages and are also valuable in selecting the language to be used for programming today's complex software systems.

2 Introduction

Developers of performance computing software, develop code not just for computational processing but also for management and infrastructure tasks such as communication, processor/core allocation, task management, job scheduling, fault detection, fault handling, and logging.To perform the performance analysis, we have used 3 algorithms Matrix Multiplication, Merge Sort and Sum of Array in 3 different languages C++, Java and Python. The algorithm is designed using multi-threading techniques. The performance analysis on the basis of execution time is evaluated by taking into consideration the difference between starting time and end time of execution in seconds. The aim of the project is to answer the following question: Which programming language out of C++,Python and Java is best suited for developing multi-threaded applications?

3 Concurrency in C++

The C++ library contains the C library as well as a class to manipulate string of characters and the Standard Template Library containing containers or data

structures such as algorithms, an improved String library and input/output stream libraries.For C++ programming we are using ¡pthread.h¿ library to use the multiple threads.Also, we are using ¡time.h¿ library to evaluate the execution time of the program.

4 Concurrency in Java

Java was designed by Sun Microsystems (now Oracle) starting in the early 1990's as a way to develop software for small embedded devices that were connected to each other.Java contains package, class, file, variable naming conventions that have been well accepted and generally implemented by the developer community.We are using the Java threads as show in the below example:

```
public class Main extends Thread
public void run()
System.out.println("This code is running in a thread");
```

5 Concurrency in Python

In Python, the things that are occurring simultaneously are called by different names (thread, task, process) but at a high level, they all refer to a sequence of instructions that run in order.In Python Threading and asyncio both run on a single processor and therefore only run one at a time. They just cleverly find ways to take turns to speed up the overall process. Even though they don't run different trains of thought simultaneously, we still call this concurrency.With multiprocessing, Python creates new processes. A process here can be thought of as almost a completely different program, though technically they're usually defined as a collection of resources where the resources include memory, file handles and things like that. One way to think about it is that each process runs in its own Python interpreter.In this project we are using "Threading" in all our programs for Concurrency.

6 Benchmarks

6.1 Matrix Multiplication

We measure the performance of matrix multiplication to examine how each language's concurrency framework handles shared data structures between threads. The benchmark multiplies a matrix M1 by another matrix M2, where M1 and M2 are the matrix of size provided of unsigned integers. We have designed 3 different programs each in C++, Java and Python for Matrix Multiplication. The programs are tested with different numbers of threads and by changing the value of N(size of Matrix).The performance of the programs are compared on the basis of time required for the execution in seconds.

	C++	Java	Python
40	0.007	0.005	0.136
80	0.003	0.004	0.808
120	0.008	0.004	2.53
200	0.039	0.005	11.82
400	0.35	0.004	134.75
800	3.27	0.002	302.23
1200	15.85	0.004	714.54
1600	44.42	0.004	911.07
2000	88.74	0.005	1289.98

Figure 1: Matrix Multiplication

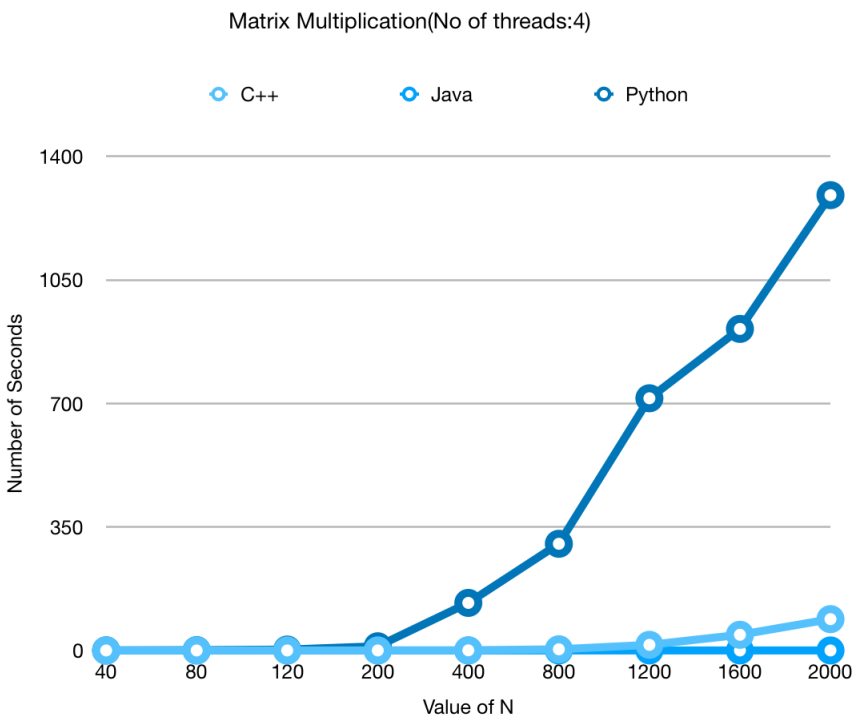

Figure 2: Matrix Multiplication Graph

For comparison we have used 4 threads and the values of N ranging from 40 to 2000. As you can see in the above graph the time taken by C and Java is almost similar.While as the value of N increases the graph of Python increases, it takes 302.23 for a Python program to compute a Matrix Multiplication of M[800][800]. While the execution time in Java is least 0.002 and the time in C++ is 3.27. So, we for Matrix Multiplication algorithm Java turns out to be the most efficient language when it comes to multi-threading.

6.2 Sum of Array

We measure the performance of Sum of array to examine how each language's concurrency framework handles shared data structures between threads.For this we will take different number of threads and will continuously change the value of size of Array to compute the efficiency. We have designed 3 different programs each in C++, Java and Python for Sum of Array.The performance of the programs are compared on the basis of time required for the execution in seconds.

	C++	Java	Python
100	0.000272	0.001	0.0177
500	0.000333	0.001	0.024
1000	0.000326	0.002	0.0210
5000	0.000285	0.001	0.0193
10000	0.000371	0.002	0.023
15000	0.000509	0.001	0.021
20000	0.00060	0.002	0.024
50000	0.00090	0.002	0.028

Figure 3: Array Sum

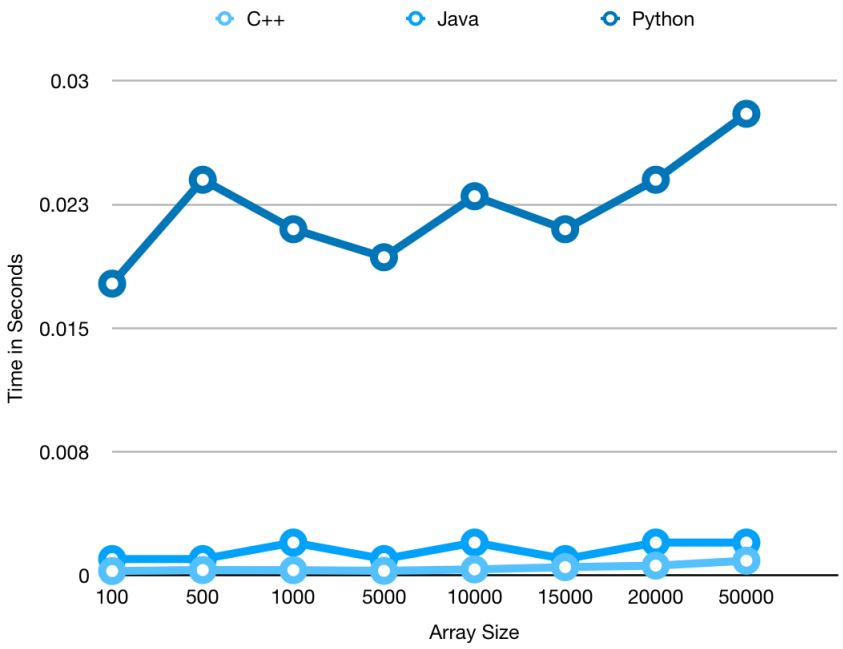

Figure 4: Array Sum Graph

For comparisons we have taken 5 threads and the value of Array size ranging from 100 to 50000. As you can see in the above graph the time taken by C++ is the minimum as compared to Java and Python. Also, the time taken by Python is the maximum, for array size it 15000 Python takes 0.021 seconds to execute while C++ takes 0.000509 seconds. While Java shows a good average performance better than Python.

6.3 Merge Sort

We measure the performance of Merge Sort to examine how each language's concurrency framework handles shared data structures between threads.For this we will take different number of threads and will continuously change the value of size of Array to compute the efficiency. We have designed 3 different programs each in C++, Java and Python for Merge Sort.The performance of the programs are compared on the basis of time required for the execution in seconds.

	C++	Java	Python
100	0.00043	0.003	0.537
500	0.0005	0.003	0.529
1000	0.0008	0.003	0.535
5000	0.003	0.010	0.5277
10000	0.005	0.015	0.5261
15000	0.009	0.028	0.5187
20000	0.011	0.028	0.5194
50000	0.029	0.034	0.5131

Figure 5: Merge Sort

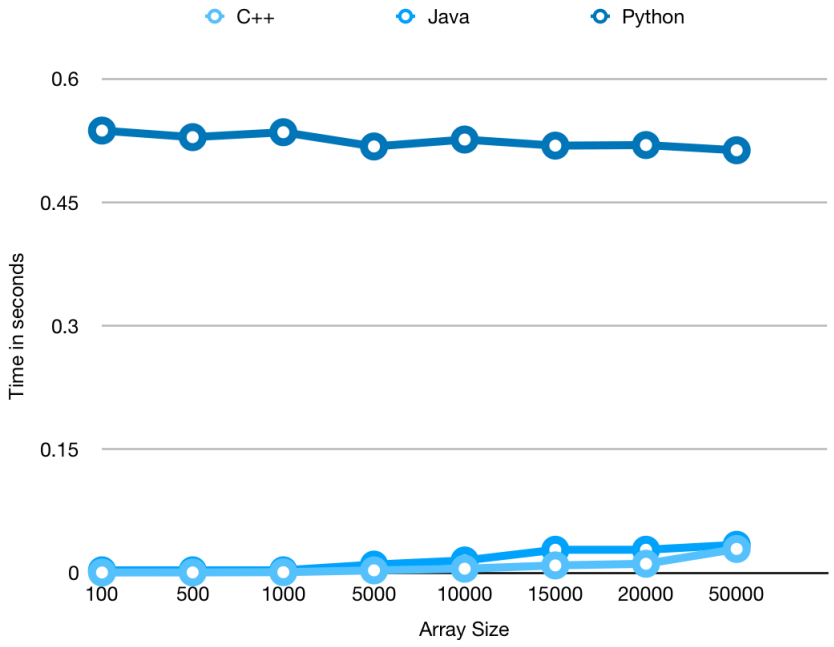

Figure 6: Merge Sort Graph

For comparisons we have taken 4 threads and the value of Array size ranging from 100 to 50000. As you can see in the above graph the time taken by C++ is

the minimum , also the execution time by Java is very less.Again Python turns out to be costlier, it take much more time for a python program to execution the multi-threading programming.

However, it is possible that the same program, written in different languages, or running under different operating systems, may exhibit significant differences in speed and efficiency.

7 Table

Language list with respective compiler or interpreter name and version:

Language	Compiler/interpreter	Version
C++	GNU g++	4.8
Java	Sun JDK javac/java	13.0.1
Python	Python	3.9.0

Figure 7: Version

8 Future Scope

Future work could for instance test more complicated multi-threaded programs that have greater risks of deadlocks.We can compare more languages such as Rust, GO, Rest for multi threading performance.

9 Conclusion

In our multi-threaded assessment, C++ and Java was measurably faster than Python. The performance differences between C++ and Java were very small.While python requires much more time to compute the execution each of the algorithm,python prevents one thread from executing at a time, preventing concurrency, leading us to the conclusion that C++ is the best suited language for developing a multi-threaded key-value store. Also, when it comes to memory management in Python and Java we don't have to worry about how and when the memory is cleared. Python and Java automatically handles the memory using its garbage collector.This collectors does not exist in C/C++. This benchmark provides a comparison of three commonly used programming languages under single operating system. The overall comparison shows that a

developer should choose an appropriate language carefully, taking into account the performance expected and the library availability for each language.

[1] [2] [4] [3]

References

[1] LOVE BRANDEFELT and HUGO HEYMAN. *A Comparison of Performance Implementation Complexity of Multithreaded Applications in Rust, Java and C++*. 2020.

[2] Josh Pfosi, Riley Wood, and Henry Zhou. *A Comparison of Concurrency in Rust and C*.

[3] Martin Rinard. *Analysis of Multithreaded Programs*.

[4] Andrej Zentner and Robert Kudelić. *Multithreading in .Net and Java: A Reality Check*. 2017.