

Comparing Programming Languages:

A Study of the Performance of Sorting Algorithms

Tobias Zeier, the 16th of June 2024
University of Essex Online

Introduction

In this essay, the student will compare three different programming languages. To do this, he analyses the runtimes of different sorting algorithms, sorting a text file with over 830,000 values. The scripts will be run on an Apple MacBook Air M3, 4.05 GHz with 8GB of memory and a connected power supply. The C and PowerShell applications will be run from Visual Studio Code (v. 1.89.1 and clang-1500.3.9.4) and the Python applications from PyCharm Community Edition (v. 2024.1.1). Since the sorting algorithm will use compute power to calculate the task, the execution times will vary between different computer models and their installed hardware. In the given scenario, the student is using the CPU for calculation, however Ye et al. (2010) found that the performance can be improved by using GPU calculation.

Sorting Algorithms

There are many different sorting algorithms which can be used for different purposes. Each of these algorithms has its own special features as well as strengths and weaknesses. They differ in the way in which the input values are sorted. For each algorithm, the three values best, average and worst are specified. These can be used to calculate how many iterations must be run through until the input values are sorted. Bubble Sort, for example, one of the oldest sorting algorithms, starts with the first input value and compares it with the second. If the second value is lower than the first, the two values are swapped. It then continues by comparing the second and third values. The iteration continues until the algorithm has reached the last value. After that, the second iteration starts, whereby the same principle is run through a second time. There are therefore as many iterations until no more values need to be swapped. The longer the list of input values, the worse the sorting performance gets, since the number of iterations increases. This results in the best-case scenario where all input values are already sorted, to the average and worst case where the number of inputs squared (n^2) need to be processed (Al-Kharabsheh, et al, 2013).

Merge sort divides the list into two equal halves, sorts them recursively, and then merges them back together to create a sorted list. It has a best, average and worst complexity of $O(n \log n)$ and typically uses three arrays. It is faster than heap sort for larger sets but requires more memory. Quick sort is usually a better choice, especially for machines with limited memory. Like quick sort, merge sort is a divide and conquer algorithm with a partition and merge phase. In the merge phase, the two sorted halves are merged by comparing and arranging values. Unlike quick sort, merge sort always partitions the list in half, resulting in an $O(n \log n)$ algorithm (Pandey, 2008).

Another factor that needs to be considered, is whether the algorithm supports stable sorting or not. Stable sorting is, when the algorithm maintains the relative order of records with equal keys (Al-Kharabsheh, et al, 2013).

Runtime Comparison

In the following table, the student notes the runtimes for the selected sorting algorithms. They are all run three times to calculate an average runtime.

Language	Algorithm	Runtime in Seconds			
		1 st run	2 nd run	3 rd run	Average
C	Bubble Sort	1263.63	1242.94	1231.04	1245.87
Python	Bubble Sort	31166.40	31633.72	30913.51	31237.87
Python	Merge Sort	2.2874	2.2772	2.2879	2.2841
Python	Tim Sort	0.1506	0.1471	0.1414	0.1463
Python	Heap Sort	0.5060	0.4688	0.5024	0.4924
Python	Quick Sort	1.5402	1.5483	1.5432	1.5439
Python	Built-in (Tim Sort)	0.3507	0.3493	0.3471	0.3490
PowerShell	Built-in	17.6054	16.3272	16.8786	16.9370

Table 1: Comparison of Execution Time in differen Programming Languages

Comparing the two languages by analysing the runtime of a bubble sort algorithm, it can be observed that the compiled language, in this example C, is much faster than an interpreted language, such as Python. The reason for this is that the compiled language is translated into machine code with the help of a compiler before the programme is executed. The interpreted language, however, is interpreted during the execution. Therefore, it takes more time for a compiled language before it can be executed, however it is faster during the actual execution (Ernest, Mensah and Gilbert, 2017).

Furthermore, it is interesting to see, that Timsort is faster compared to Python's built-in sort, which is also based on Timsort. This could be due to Python's implementation of Timsort but also due to the given data set.

Hashing Function

As an additional task the student is asked to create a hash from a sorted file, then modify the text file by entering his name, date of birth and student number and again create a hash from the modified file. The two hashes are then compared. For this task, the student uses the Secure Hash Algorithm-3 (SHA3-256) which is the latest version.

In their groundbreaking 1976 paper on public-key cryptography, Diffie and Hellman recognised the importance of one-way hash functions in creating digital signature schemes. Cryptographic hash functions can take input strings of any length and produce short, fixed-length output strings. The concept and designs for these hash functions were first introduced in the late 1970s by Rabin, Yuval, and Merkle (Preneel, 2009).

The algorithm operates by extracting content from a given file and producing a unique hash based on that content. It can be observed that any modifications made to the file, such as appending the author's name, date of birth, and student number, will result in a change to the generated hash value. The first hash below is generated from the original file, the second one from the modified file.

SHA-3 hash of pidata.txt:

f924005c0701d2e15fffaadadbba7e4e21663822dd230c5ba07322159acf0d83

SHA-3 hash of pidata2.txt:

31dcbd73822e6675f6e4945e2d8523feb6def7fdee5da467bf22e2a78106f1c0

Students Experience with Python

The student has a strong background in systems engineering, working with Windows Servers. Therefore, he is used to write PowerShell scripts to automate specific tasks which is also the reason why he added PowerShell as a third programming language to the task in this assignment. In his experience Python is a relatively easy to learn programming language. The variety of libraries makes it powerful, and the indentation helps reading code and provides a coherent structure. This assignment was the students first interaction with the programming language C, which he experienced as rather complex since it needs many lines of code and formulations.

Python is only used for specific use cases in the organisation where the student works, as the team of Enterprise IT Architects has implemented a governance policy that primarily allows the use of Java and Kotlin as programming languages.

Word count: 1086 Words

List of Tables:

Table 1: Comparison of Execution Time in differen Programming Languages..... 2

References:

Ye, X., Fan, D., Lin, W., Yuan, N. & lenne, P. (2010) High performance comparison-based sorting algorithm on many-core GPUs. *IEEE International Symposium on Parallel & Distributed Processing (IPDPS)*: 1-10. DOI:

<https://doi.org/10.1109/IPDPS.2010.5470445>

Al-Kharabsheh, K. S., AlTurani, I. M., AlTurani, A. M. I., & Zanoon, N. I. (2013) Review on sorting algorithms a comparative study. *International Journal of Computer Science and Security (IJCSS)*, 7(3): 120-126.

Pandey, R. C. (2008) *Study and Comparison of various sorting algorithms*. Master thesis. Thapar University. Available from: <http://hdl.handle.net/10266/583> [Accessed 2 June 2024].

Ernest, A., Mensah, E. & Gilbert, A. (2017) Qualitative Assessment of Compiled, Interpreted and Hybrid Programming Languages. *Communications on Applied Electronics* 7(7): 8-13. DOI: <https://doi.org/10.5120/cae2017652685>.

Preneel, B. (2009). The State of Hash Functions and the NIST SHA-3 Competition. *Lecture notes in computer science*: 1–11. DOI: https://doi.org/10.1007/978-3-642-01440-6_1