OPEN ACCESS Freely available online

Journal of Theoretical and Computational Science

Perspective

Algorithms Use in Sorting Technology

Huimin Weibo*

Department of Electronic Information and Automation, Civil Aviation University of China, Tianjin, China

DESCRIPTION

Around half of all computing time is spent searching and sorting on computers around the world. Scientific computing, image processing, multimedia and network processing, database operations, radio networks, artificial intelligence and robotics, scheduling, data compression, and scientific computing are only a few examples of applications that use sorting. The Big-O notation is used to express the algorithmic complexity of the sorting method, where O indicates the difficulty of the algorithm and n denotes the set's input size. O (n2), for example, denotes quadratic complexity in an algorithm.

Comparisons are used in all compareson-based sorting algorithms to establish the relative order of elements. Insertion sort, merge sort, fast sort, and heap sort are some examples. For serial algorithms, the best worst-case running time for comparison sorting is O (n log). Quick sort, merge sort, tim sort, intro sort, heap sort, in-place merge sort, insertion sort, selection sort, block sort, cube sort, exchange sort, shell sort, bubble sort, tree sort, cycle sort, patience sorting, smooth sort, library sort, strand sort, tournament sort, cocktail shaker sort, comb sort, gnome sort, and odd-even sort are just a few examples of well-known comparison sorts.

Some comparison-based algorithms can be implemented on parallel computer architectures, with the worst-case complexity of such parallel algorithms being O (n).

Non-comparison sorts do not rely on element comparison to establish order, and they are conceivable in linear time (under certain assumptions), i.e. O (n). As a result, they aren't restricted to O (n log n). Many of them are predicated on the notion that the key size is large enough to ensure that each entry has a unique key value. Pigeonhole sort, bucket sort (uniform keys), bucket sort (integer keys), counting sort, LSD radix sort, MSD radix sort, MSD radix sort (in-place), spread sort, burst sort, flash sort, and postman sort are among the most well-known non-comparison sorts. By efficiently spreading data into numerous buckets and then sending down sorting to several processors, sample sort can be used to parallelize any of the non-comparison sorts, with no requirement to merge buckets that are already sorted between each other. Whether the sorted output is achieved using comparison sort, non-comparison sort, or a hybrid sorting method, parallel computing can be used to speed up sorting.

Each sorting algorithm has its own set of benefits and drawbacks, but regardless of the algorithm, the hardware components of the processor that are employed to produce sorted output are ALUs (Arithmetic and Logic Unit). The main processes used in the sorting of data lists for comparison-based algorithms are comparison and swapping. Internally, processors such as the Intel 8085 and 8086 series use a digital circuit called a subtractor, which is present in the ALU, to conduct comparison operations. Because it takes the execution of specialized instructions to examine the flag registers to confirm the comparison findings, this comparison is slower. A comparator circuit is a specialized digital circuit that can be utilized for comparison operations. Comparators do not employ subtraction operations (subtractor) to do comparisons, instead relying on internal logic circuitry to do so.

It is commonly known that array data structures are contagious memory clocks that are used to store input in comparison or non-comparison based sorting. The addressing computation of array elements to carry out read and write operations is involved in the rearranging of array elements in an unsorted array. ALU's digital circuits, known as adders and multipliers, are used to compute the addresses of array elements using the initial address of the array's first element. The speed of the addition operation is largely influenced by whether the digital adders are serial or parallel. Non-comparison algorithms, such as radix sort, use digital arithmetic circuits to compute the next significant digit necessary at each pass. This can also be accomplished using arithmetic operations conducted in an ALU. The sorting process is directly related to the design of arithmetic circuits and comparators, as well as the digital technologies utilized to speed up the operations of these circuits.

Because the components of ALU adders, subtractors, and comparators are directly involved in the sorting process, the structure and architecture of ALU influence the sorting. Again, serial versus parallel processing has a significant impact on the

Correspondence to: Huimin Weibo, Department of Electronic Information and Automation, Civil Aviation University of China, Tianjin, China, E-mail: huimin.weibo@126.com

Received: 21-Apr-2022, Manuscript No. JTCO-22-17866; **Editor assigned:** 26-Apr-2022, PreQC No. JTCO-22-17866 (PQ); **Reviewed:** 10-May-2022, QC No. JTCO-22-17866; **Revised:** 17-May-2022, Manuscript No. JTCO-22-17866 (R); **Published:** 25-May-2022, DOI: 10.35248/2376-130X.22.8.147

Citation: Weibo H (2022) Algorithms Use in Sorting Technology. J Theor Comput Sci. 8:147

time it takes to sort a given input. I believe there is a lot of potential for speeding up the sorting process by inventing specialized hardware to solve this challenge, because the hardware is the one that has to do the sorting internally. Comparator and Swap Circuitry (CAS) or non-comparison based hardware techniques can be used for sorting. Specialized instructions can also be added to the processor's instruction set.

CONCLUSION

The findings of this study shows that there is a lot of room for improving application-specific sorting technology that can sort faster and without using a comparator. There is still a need to revisit this topic and consider non-traditional ways in order to achieve efficient speed and memory results. Moving from O (n2) to more desirable O (n log n) solutions in comparison-based algorithms, for example, is not the only technique to make the sorting process more efficient. Using innovative hardware circuit designs, the process can be improved even further.