Enhancing the Radix Sort using Clustering:

Sorting is a method of arranging data in a certain order, let it be ascending, descending, or lexicographical. The goal of the paper is to use a clustering technique in conjunction with radix sorting. On the basis of their distance from one other, the elements to be sorted are divided into several clusters. The K-mean clustering technique is used for this. Then, using radix sort, these clusters are sorted independently. The efficiency of radix sort has been improved by utilizing this strategy. This work is based on a review of several research publications in the area of Radix Sort.

Given that the date stored in the database is in well sorted order, the data searching could be optimized to very high level.

For comparisons and temporary storage of a few data components, sorting algorithms may require some extra capacity. In-place sorting algorithms are sorting algorithms that do not require any extra space for sorting and sorting is said to happen in-place, or in other words, within the array itself. Bubble sort is an example of in-place sorting. This applies to In-place sorting and Not In-place sorting. Similarly, the sorting can be further divided into Stable and Not Stable sorting algorithms, Adaptive and Not-Adaptive sorting algorithms.

Radix Sort:

Radix Sort is a non-comparison-based integer sorting method that groups integer keys by individual digits that have the same significant position and value. Radix sort is not restricted to integers since integers can represent strings of characters (e.g., names or dates) as well as specifically prepared floating-point numbers. Heterogeneous databases comprise numbers, strings, and floats, and MRS is a radix sort method created by Avinash Shukla and Anil Kishore Saxena for sorting heterogeneous data sets.

There are two types of radix sort, LSD Radix Sort and MSD Radix Sort respectively. LSD (Least significant digit) Radix sort is in which the process starts from least significant digit to most significant digit and MSD (Most significant digit) Radix sort is in which the process starts from most significant digit to least significant digit.

Coming to the efficiency part, the complexity of a radix sort is O(wn), where n is the number of keys to sort and w is the key's word size, or the number of digits. When w is written as a constant, radix sort outperforms best comparison-based sorting algorithms (for sufficiently big n), which all take O(n log n) comparisons to sort n number of keys. However, w cannot be regarded a constant in general since if all n keys are unique, w must be at least log n in order for a random-access computer to retain them in memory.

Clustering:

Our goal is to improve the performance of the current Radix Sort algorithm in one of two ways:

1) By modifying hardware

2)By modifying a pre-existing algorithm

The following are the components of hardware customization:

1) Multi-core CPUs are being added.

2) Primary Memory Addition

3) Cache has been added.

Using a clustering method, we will customize an existing algorithm.

1) Creating the data clusters that will be sorted.

2) Data might be grouped at the nearest memory location.

3) Scheduled processes can be bundled and processed in batches this manner.

4) Batch processing would improve performance since the CPU's latency time would be reduced. We shall organize jobs into clusters for batch processing in our work.

Mean is calculated for all the clusters and the elements are rearranged in the clusters based on the distance of them from the mean. The elements in the data set are to be sorted and divided into clusters, and each cluster is given its own Radix Sort. As a result of this procedure, the time required for sorting will be reduced, and the efficiency of the radix sort will be improved.

To conclude, after having a close look into the literature part of the paper, it comes out be that Recursive MSD Radix sort turns out be the best sorting algorithm with time efficiency less than n square. When sorting strings in a huge data collection, Recursive MSD Radix Sort is the most efficient algorithm. After reviewing the existing research, it has been determined that the two most essential aspects in sorting algorithms are time complexity and space complexity. Radix Sort has been implemented in a variety of ways, all of which lower the time and space complexity of the algorithm.

Research paper took for reference:

https://issuu.com/ijsrd/docs/ijsrdv4i50417