Sorting of Data in Consumptional Model Using Big Data Technique

[1]D. Dhivya, [2]Dr. K. Nirmala
[1]M. Phil Research Scholar, [2]Associate Professor
[1]PG & Research Dept. of Computer Science,
[1] Quaid-E-Millath Government College for Women (Autonomous), Anna Salai, Chennai 600002, Tamil Nadu, India

Abstract : One of the major problem in computer science is arranging the data in the database. The different types of sorting algorithms are used to overcome this kind of issues and it helps to arrange the data of a list in a certain order. It makes simple access for searching and locating the information. The mostly used orders to arrange the data is numerical and lexicographical order. The efficiency of the sorting algorithms is to optimize the importance of other sorting algorithms. The optimality of such sorting algorithms is evaluated while calculating their time and space complexities. An efficient sorting algorithm uses less time and space to process large amount of data. In this regard the work implements MapReduce algorithm to reduce the space and time complexity over sorting.

IndexTerms - **Sorting, MapReduce algorithm, Time complexity and Space complexity**

I. INTRODUCTION

Sorting is arranging the list of data. We can differentiate sorting in two ways. If the size of data is small and fits in to main memory, then the sorting is called internal sorting and if the size of data is large and some of them are resides in external storage during the sort, then it is called external sorting. Sorting has been a significant zone for the algorithmic analysts and numerous assets are contributed to recommend more works for sorting calculations [2]. This work tried to sort out the dataset of Earthquake measurement with the attributes like source, id, date and time, latitude, longitude, magnitude, depth, NST and region. The utilization of space and time were calculated according to the different size of dataset.

II. LITERATURE REVIEW

Vignesh R, Tribikram Pradhan [1], in 2016, presented a new approach merge sort enhanced in place sorting algorithm. This paper discusses about experimental analysis of worst case, best case and average case and noticed that the code slows down for very large values of 'n'. The instability of this algorithm is also a cause of concern.

Sehrish Munawar Cheema, Nadeem Sarwar, Fathima Yousaf [2], in 2016, Presented a new approach contrastive analysis of bubble & merge sort proposing hybrid approach. In this paper the merits and demerits of two sorting techniques are discussed. The bubble sort is better than merge sort in practice for small set of data, but as size of input data increases, the performance of bubble sort suddenly drops down. Keeping in view the previous flaws in merge and bubble sort algorithms Hybrid Approach can be used to eliminate flaws.

Shrinu Kushagra, Alejandro Lopez-Ortiz and J. IanMunro [3], in 2013, presented a new approach which consisted of multiple pivots in order to sort elements. They performed an experimental study and also provided analysis on cache behavior of these algorithms. Here, they proposed a 3 pivot mechanism for sorting and improved the performance by 7-8%.

Rohit Yadav, Kratika Varshney and Nitin Verma[4],in the year 2013 discussed the run time complexities of the recursive and non recursive approach of the merge sort algorithm using a simple unit cost model. New implementations the for two way and four way bottom-up merge sort were given, the worst case complexities of which were shown to be bounded by 5.5n log 2n + O(n) and 3.25n log 2n + O(n), respectively.

You Ying, Ping You and Yan Gan[5], in the year 2011 made a comparison between the 5 major types of sorting algorithms. They came to a conclusion that Insertion or Selection sort performs well for small range of elements. It was also noted that Bubble or Insertion sort should be preferred for ordered set of elements. Finally, for large random input parameters, Quick or Merge sort outperforms other sorting algorithms.

Wang Xiang[6], in the year 2011 presented a brief analysis of the performance measure of Quick Sort algorithm. This paper discusses about the Time Complexity of Quick Sort algorithm and makes a comparison between the improved Bubble Sort and Quick Sort through analysing the first order derivative of the function that is found to co-relate Quick Sort with other sorting algorithms.

III. METHOD: MAPREDUCE ALGORITHM

MapReduce algorithm is used to process tremendous amount of data in parallel, reliable and efficient way in cluster environments. It applies Divide and Conquer technique to process large amount of data. The data input splits into smaller and manageable parts and then executed in-parallel way.

MapReduce Algorithm Steps:

This algorithm consists of three following steps they are:

1. Map Function
2. Shuffle Function
3. Reduce Function

MAP FUNCTION:

The first step in MapReduce Algorithm. It takes the input data and tokenize it into smaller sub data then performs the required task on each sub-data in parallel. It further performs the following two sub-steps they are:

1. Splitting
2. Mapping

- ➢ Splitting step takes input Dataset from the Source and tokenize it into smaller Sub-Datasets.
- ➢ Mapping step takes those smaller Sub-Datasets and perform required action or computation on each Sub-Dataset

The final output of this Map Function contains a set of key and value pairs as <Key, Value>

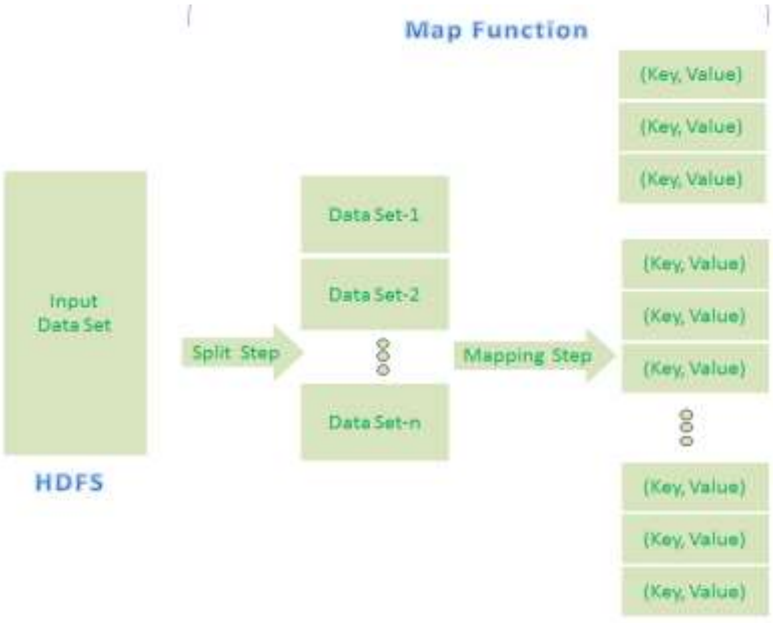

Fig 1: Mapping Function

SHUFFLE FUNCTION:

The second step in MapReduce Algorithm also known as “Combine Function”. It also further performs the following two sub-steps:

1. Merging
2. Sorting

It takes the list of output from “Map Function” and perform these two sub-steps on each and every key-value pair.

- ➢ Merging step combines all key-value pairs which have same keys. This step returns <Key, List<Value>>.

- Sorting step takes input from Merging step and sort all key-value pairs by using Keys. This step also returns <Key, List<Value>> output but with sorted key-value pairs. Finally, Shuffle Function returns a list of <Key, List<Value>> sorted pairs to next step.

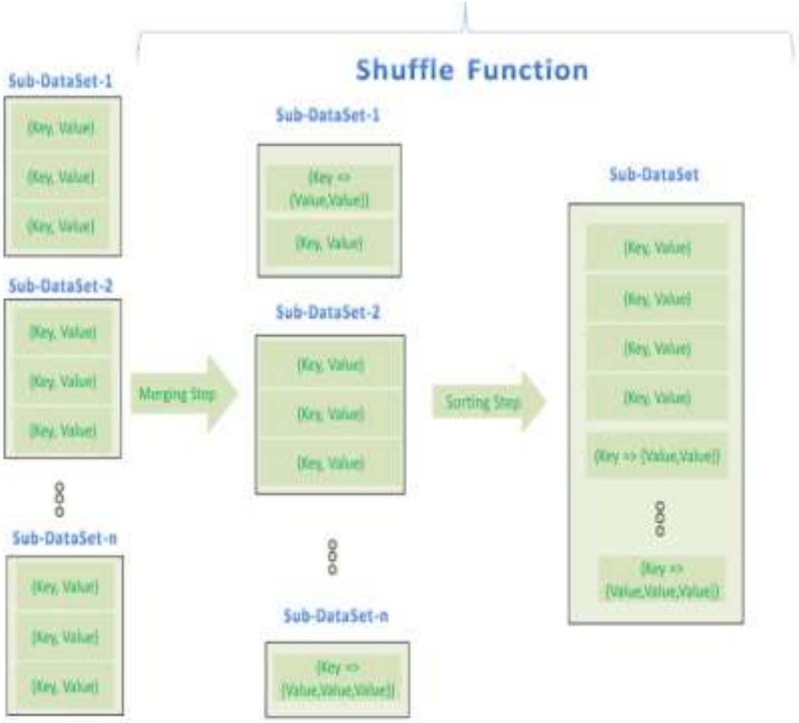

Fig 2: Shuffle Function

REDUCE FUNCTION:

The final step in MapReduce Algorithm performs the Reduce step. It takes the list of <Key, List<Value>> from Shuffle Function and perform reduce operation.

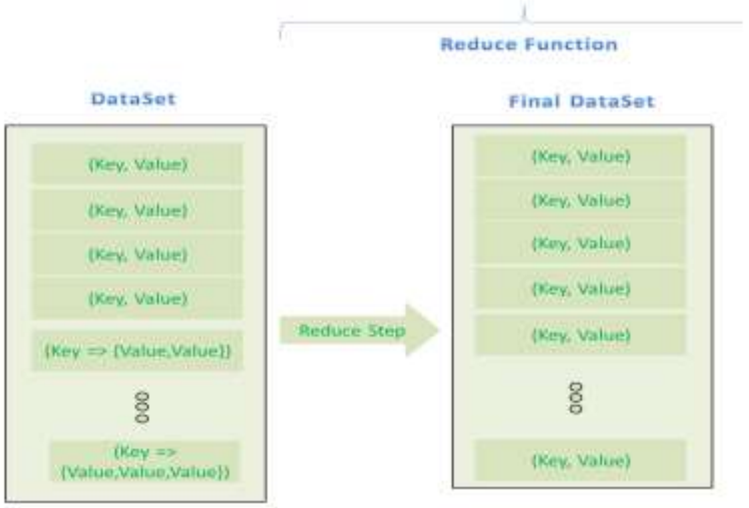

Fig 3: Reduce Function

IV. RESULTS AND DISCUSSION:

Time Utilization:

Utilization of time is based on size of the dataset. According to the different size the time is increased and decreased.

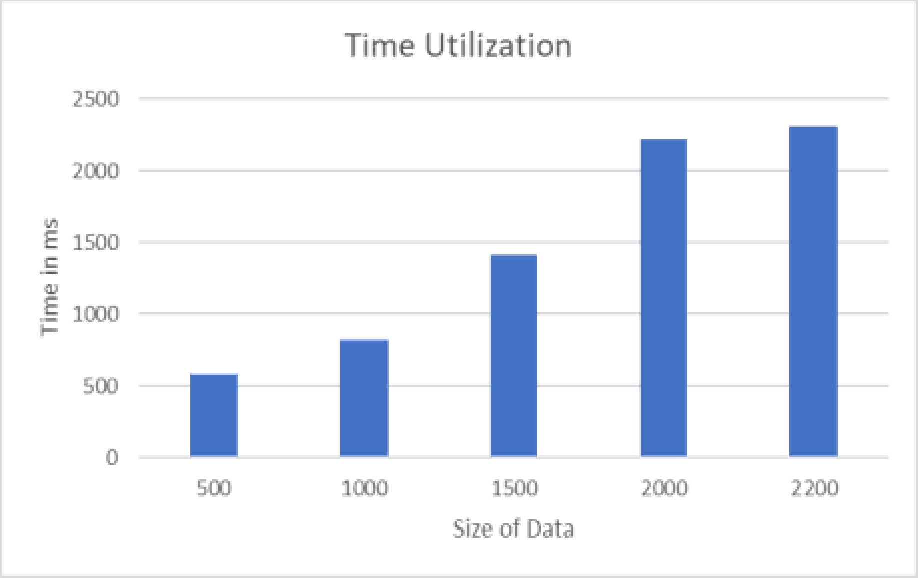

Fig 4: Utilization of Time

Memory Utilization:

The memory is utilized by different size of dataset and it has been measured in bytes.

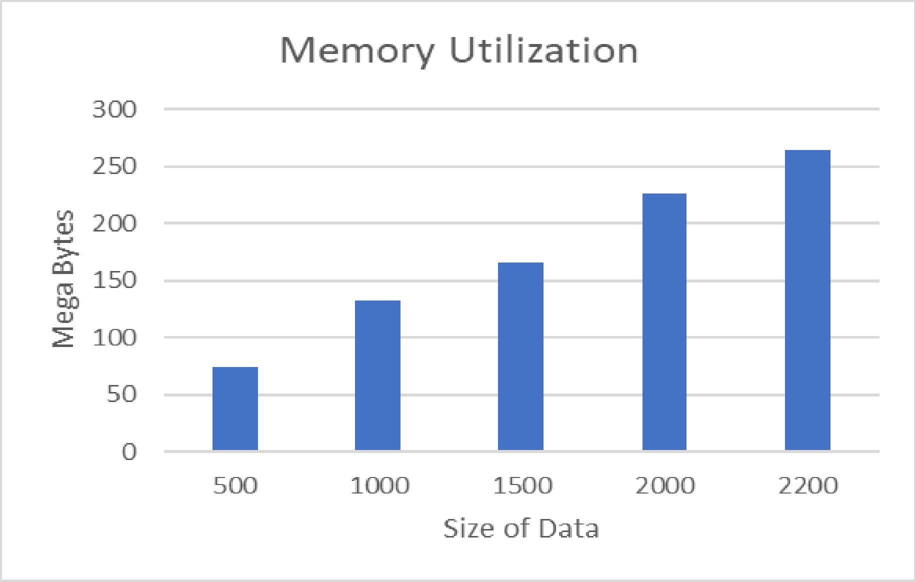

Fig 5: Utilization of Memory

V.CONCLUSION:

Sorting approach was implemented using MapReduce algorithm to reduce the number of comparisons, space and time complexities among the data. MapReduce provides a distributed parallel computing across multiple nodes and return result on a particular node. MapReduce plays a vital role in parallel data processing because of its salient features such as scalability, flexibility and fault tolerance.

VI. REFERENCES:

[1]. Vignesh R, Tribikram Pradhan. “Merge Sort Enhanced In Place Sorting Algorithm”, International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), 2016

[2]. Sehrish Munawar Cheema, Nadeem Sarwar, Fathima Yousaf. “Contrastive Analysis of Bubble & Merge Sort Proposing Hybrid Approach”, Sixth International Conference on Innovative Computing Technology (INTECH 2016)

[3]. Shrinu Kushagra, Alejandro Lopez, J. Ian Munro and Aurick Qiao “Multi-Pivot Quicksort: Theory and Experiments”, Proceedings of the 16th Meeting on Algorithm Engineering and
Experiments (ALENEX), pp. 47-60, 2014.

[4]. Rohit Yadav, Kratika Varshney and Nitin Verma. “Analysis of Recursive and Non Recursive Merge Sort Algorithm”, International Journal of Advanced Research in Computer Science and
Software Engineering Volume 3, Issue 11, November 2013

[5]. You Yang, Ping Yu and Yan Gan. “Experimental Study on the Five Sort Algorithms”, International Conference on Mechanic Automation and Control Engineering (MACE), 2011

[6]. Wang Xiang. “Analysis of the Time Complexity of Quick Sort Algo- rithm”, Information Management, Innovation Management and Indus- trial Engineering (ICIII), International
Conference, 2011