INFO 6205 Program Structures & Algorithms Fall 2021 Final Project - Literature Review

R C HILLYARD*, College of Engineering, Northeastern University, USA
Kaeyang Hsieh (001092745), College of Engineering, Northeastern Universty, USA
Nienchi Hung (002980433), College of Engineering, Northeastern Universty, USA
Shrusti Rajesh Chheda (002196756), College of Engineering, Northeastern Universty, USA

In computer science, sorting is usually classified into comparison-based and non-comparison-based algorithms. Radix sort is an efficient algorithm to sort strings among the non-comparison-based algorithms. Here we have discussed the work of three research papers that talk about improvement to the traditional Radix sort using Parallel Radix sorting and algorithmic caching. The cache-efficiency improvement is implemented in this final project.

1 INTRODUCTION

Sorting is a fundamental technique in computer science. It has been applied to numeric infrastructures like databases, computational tasks, network devices, which play critical roles in human life. Sorting strings is one of the most common tasks in daily life, and MSD (Most Significant Digit first) Radix sort is the best-known and the most efficient algorithm to sort strings. Sorting involves array accesses, comparisons, and swapping operations. Hence the running time complexity is affected by parameters such as memory access patterns and CPU utilization. There has been extensive research to improve the string sorting algorithms. In this paper, we attempt to understand some methods that can accelerate the MSD radix sorting speed.

2 REVIEW OF LITERATURE

2. Increase cache-efficiency

The original MSD radix sort accesses memory irregularly, which violates the design for modern CPU architecture. The paper[1] increases cache memory utilization by using loop fission(CE2) and replacing the random array access with sequential array access(CE1).

2.2 Parallel Sort

Modern CPUs have multiple cores architecture, which can be leveraged to speed up MSD radix sort by paralleling the algorithm. The first one is PARADIS[2], which repeats the permutation and repair phases. The permutation phase distributes the keys equally to buckets that are assigned to an individual processor, respectively. The repair phase is to repair the keys that may be distributed to the wrong bucket in the permutation phase. The other work is Region sort[3], which stores keys of the bucket in a graph, where the directed edge is a set of misplaced keys and a vertex represents a country. The sorting procedure swaps the misplaced keys parallelly and recursively to a correct place until no edges exist.

The benchmark result shows that the region sort achieved a speedup around 19-65x on a 36 cores machine with hyper-threading compared to the other state-of-the-art radix sort and out-of-place radix sort algorithm. As the algorithm, PARADIS[2], has no public code, we cannot compare them.

3 DISCUSSIONS

We implemented the techniques to increase cache efficiency; however, there is no improvement but deterioration(the statistic is in the report). We believe that the technique might not be applied on Java, while the original improvement was applied on C/C++. Another reason is that the compiler design has improved a lot since 2008, thus diminishing the need to arrange the code for better cache efficiency.

4 CONCLUSION

The paper summarized two parallel sorts and a cache-efficiency improvement method. However, implementation of the cache-efficiency improvement approach to our sorting method does not match our expectations. The parallel sort is too complicated and will need much more time to implement.

REFERENCES

[1] Kärkkäinen, J., & Rantala, T. (2008, November). Engineering radix sort for strings. In International Symposium on String Processing and Information Retrieval (pp. 3-14). Springer, Berlin, Heidelberg.

[2] Cho, M., Brand, D., Bordawekar, R., Finkler, U., Kulandaisamy, V., & Puri, R. (2015). PARADIS: an efficient parallel algorithm for in-place radix sort. Proceedings of the VLDB Endowment, 8(12), 1518-1529. Chicago

[3] Obeya, O., Kahssay, E., Fan, E., & Shun, J. (2019, June). Theoretically-efficient and practical parallel in-place radix sorting. In The 31st ACM Symposium on Parallelism in Algorithms and Architectures (pp. 213-224).