WCIT-2010

Heuristic and pattern based Merge Sort

Manouchehr Zadahmad jafarloua*, Parisa Yousefzadeh farda

aDep. of Computer, Islamic Azad University-Ilkhchi Branch, Imam St., Ilkhchi, Iran

Abstract

The aim of this study is to present a stable and adaptable Merge Sort algorithm that uses the design patterns to reduce computational complexity of swaps and memory usage. The order of Settlement of elements recorded by design patterns and merging one element by another (first phase) replaced with chunk merging. This algorithm has been implemented by C++ programming language.

© 2010 Published by Elsevier Ltd. Selection and/or peer-review under responsibility of the Guest Editor.

Keywords: Merge sort, Design patterns, computational complexity;
2. stgEx pattern, status data structure, stgMerge algorithm and stgMerge sort

stgEx pattern is a Strategy pattern [1] based explorer pattern and has three state (three different algorithm) (Fig. 1). stgEx explore interior of under sort array. The result of each of algorithms is the data structures called status. The resulted status array of each algorithm differs in sizes, values and special viewpoints of presortedness with another one but all of them can optimize Merge sort algorithm. We call Merge algorithm the stgM and Merge sort [5,6] the stgMS, because these algorithms have three states and implemented with strategy pattern.

Fig. 1. Heuristic and pattern based Merge Sort implementation

2.1. First algorithm

In the first state, before every calling merge algorithm (stgM), the first state of stgEx pattern called and the next partial array in under sort array has found. This algorithm is a for loop with O(n) time order and produce a tow element array named status. First element of status is the index of first element of next partial array and the second one represent "the number of elements of partial array"*(+1) for ascending partial array or *(-1) for descending partial array.

For example for array -4, -3, 0, 1, 3, 8, 9, 14, 5, 6, 8, 14, 2, 1, -3, 1 ... if the (-4, -3, 0, 1, 3, 8, 9, 14) be the part of array that sorted before with stMS, (5, 6, 8, 14) will be the next partial ordered array and so the status array is (8, +4)- The result of stgEx. Then stgM called with values of status array (stgM(8, +4)) and then 0->7 indexed partial sorted array merged with 8->11 indexed new partial array and 0->11 indexed partial sorted array produced with stgMS algorithm. For the next iteration of stgMS, 0->11 indexed partial sorted array merged with (2, 1, -3) descending partial array -(stgM(12, -3))- from the last element to the first element to produce an ascending sorted array and so on.

This algorithm used when the numbers of arrays are so many that cannot move to volatile memory.

2.2. Second algorithm

The second algorithm of stgEx pattern did for whole under sort array. The result (status) is: "the first index of first entry of partial arrays" * +1 for ascending partial arrays" * -1 for descending partial arrays (O(n)). For example for array -8, -4, 0, 4, 3, 1, 0, -2, 5, 7, 9, 10, 5, 4, 2, 1, 3 status will be +1, -4, +8, -12, +15 so partial arrays are indexed 0->3 (ascending), 4->7 (descending), 8->11 (ascending), 12->14 (descending), 15->15 (ascending). Then status send as argument to stgMS and partial arrays will be the building blocks for stgM algorithm.

This algorithm used when the numbers of arrays are so many that cannot move to volatile memory.

2.3. Third algorithm
The third algorithm of stgEx pattern did for whole under sort array. The result (status) is: "the number of partial arrays" +1 for ascending arrays and "the number of partial arrays" -1 for descending arrays (O(n)). For example for array -8, -4, 0, 4, 3, 1, 0, -2, 5, 7, 9, 10, 5, 4, 2, 1, 3 status will be +3, -5, +4, -4, +1. Then status send as argument to stgMS and partial arrays will be the building blocks for stgM algorithm.

In this algorithm, stgMS is a Huffman coding algorithm [7,8,9] that optimizes merging process by choosing two partial arrays that sum of elements of them is less than others before each merging. Then these two partial arrays send as argument to stgM algorithm and merged.

Conclusion

Merge sort is an appropriate algorithm with O(n) Computational complexity, but petitioning of array to one element partial arrays and then merging them cause increasing complexity in time order, system software and hardware work. The presented algorithm eliminates these extra work using patterns.

References

1. E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995.