Abstract

One of the basic problems in computer science is sorting that need to be fast and efficient, since data is growing day by day. Various applications need fast sorting algorithms like Big Data analyses particularly in large scale scientific, social/web mining and commercial application domains. Divide and conquer Sorting Algorithms (Quick sort and merge sort) provides the best running time among all the sorting algorithms. When parallelism is applied to these algorithms, new performance leaps are accomplished. Recent parallel programming procedures and environment needs profound changes in programs to accomplish parallelism furthermore constitute puzzling, confounding and mistake inclined constructs and standards. When the number of processors utilization gets large, the overhead of thread synchronization and processor scheduling gets increase, this diminishes the speedup. In this paper, two algorithms are designed using C# viz. parallel quick sort and parallel merge sort that uses Parallel.Invoke() method. Both algorithms when executed over multicore architecture compute the threshold beyond which the above mentioned algorithms achieve speedup in comparison to its sequential version, Also threshold is calculated and compared for both the algorithms for uneven input
size.

**References**


**Index Terms**

Computer Science

Algorithms
Keywords

Parallel, threshold, multicore, speedup, sorting, complexity, processor.