Parker Loomis

CS3310

Assignment 3 Fall 2016

11/03/16

Merge-Sort and Multicore/HPC

**Problem 1: Design and analyze an efficient iterative solution to merge sort for any value of n ≥ 0.**

Step 1: Check size of list. If it is 0, return null. If it is one return that number.

Step 2: Take the length of the list size and divide by two. Take the length and also mod by 2.

Step 3: If mod result is 1, make a double value equal to the last value.

Step 4: Make a loop that divides the list into two arrays until the array is just one element.

Step 5: Make a loop that merges two arrays together. Compare every digit in the new array and sort them. Do this until there is just one single array that is sorted. During the first merge add the double element you set aside at the begin to one of the arrays.

Parker Loomis

CS3310

Assignment 3 Fall 2016

11/03/16

Merge-Sort and Multicore/HPC

**Problem 2: Design and analyze an efficient solution to sort the list A using the eight PEs efficiently using merge-sort. Again pseudo-code is sufficient but explain your design so anyone (with a reasonable programming expertise) can understand your solution easily. You may want to give a high-level description of your idea first and then give pseudo-code.**

Step 1: Check size of list. If it is 0, return null. If it is one return that number.

Step 2: Split the array into 8 equal sizes. Take the length and also mod by 8.

Step 3: If mod result isn’t 0, make a new array equal to the size of the mod result. Taking the same amount of numbers off the end of the original array and add them to the new array.

Step 4: Each core will be given an equal load that they should be able sort.

Step 5: Create a loop that merges the arrays from two of the cores into one. Then sorting them. This should loop until everything is being processed by one core.

Step 6: Add the values from the array you created in the begin with the leftover values. Sort one last time.