Luke Johnson

Lab 3

CS 555

**Synthesis Questions**

**1.      Chart the speedup, cost, and efficiency varying the sort algorithm, data set size, and number of processors.**

These charts were made in relation to P=1, N = 160,000, where the sorting function was the C qsort function.

**2.      Estimate the computation to communication ratio for a data set size of 220 and 4 processors. You can use the Big Oh estimates for processing size and can leave the answer in terms of a power of two.**

**3.      Which law (Amdahl’s or Gustafson’s) is more appropriate in this case?**

Amdahl’s

**4.      Assume that on a single core system, there is insufficient memory to hold the entire array of doubles. How could this be overcome? What modifications in the parallel version might lead to super-linear speedup should we combine a bunch of these single core systems into a cluster?**

**5.      What does the speed up results tell you about the effectiveness of parallelizing this application?**

**6.      Did any of your runs demonstrated super-linear speedup? If so, explain why? How would speedup be impacted if we ran the entire experiment using bubble sort instead of quick sort?**

**7.      Suppose the processors simply used a collective gather rather than the tree structured merge. How do you think this would affect speedup? Explain your answer.**

**8.      Do you think the parallel bitonic sorting algorithm would improve speedup compared to the odd-even approach? Justify your answer.**

**9.      What is a major advantage of partitioning approach compared to divide and conquer? Which sorting algorithm fits this model?**

You don’t end up with a single processor needing to merge all of the data at the end. The partitioning approach only requires a gather once all computations are done.

**10.  How might a shared memory implementation differ from message passing? What are some advantages and disadvantages?**

For the odd/even sort, there would necessarily be synchronization points between iterations, to prevent the shared data being corrupted between neighbors. This means there is more idle time in the loops, but it also does not require passing arrays of data between neighbors.