Matthew Irvine

Homework Assignment 2

**Q1.**

The isoefficiency of the matrix multiplication algorithm, as given by the paper, is . Breaking down the equation to individual parts we can first discuss which is the per-word transfer time. Since the per-word transfer time isn’t dependent on capabilities of the algorithm but instead on the computer that is used for the algorithm then for any particular computer it remains constant throughout the algorithm (or at least relatively constant, but it isn’t the most important part of this equation). Next is which is equal to . In this part tc is the operational cost which, much like tw, is dependent on the components of the computer and would remain relatively constant for any computer. On the other hand, E represents the efficiency of the algorithm (not isoefficiency) and we were to take the limit of we would find the result equal to 1. From this result we can conclude that large numbers of n will not heavily affect the efficiency of the algorithm. The last piece, , is our most important term because p represents the number of processors needed for the problem size W. What this indicates is that as the workload increases a rate of O(p2) is needed to maintain a particular efficiency.

In contrast, the isoefficiency of the checkboard partitioning algorithm is From this equation we can see that is the most important expression(s) as it not only decides the scalar multiple but also the exponent of p, the number of processors needed for workload W. Overall, this algorithm stays relative to O(p log p) which is better than O(p2) but depending on the parts used in the construction of the computer the exponent of p could be greater than 1 and thus perform worse than O(p log p). Worse case is that the result of is 2 or greater thus making the algorithm run worse than matrix multiplication. Because of this, when considering what machine to use the checkboard partitioning algorithm on, it is important to consider the operational costs, per-word transfer time, and overall efficiency of the computer being used before implementing this algorithm. In other words, this algorithm isn’t as scalable as the matrix multiplication algorithm because it is more dependent on the capabilities of the computer. Assuming the hypercube parallel computer has balanced communication and computational speeds then this algorithm is more scalable because the necessary number of processors for any particular workload is lower than O(p2).