

Jared Hogan

1. In the first version we are using the for clause in the pragma. This allows the work to be divide up between the threads. In the second version we are just running eight version of the same code as each thread is doing the same loop.
2. My best performance run was 15.62 GFLOPS
3.
 - a. For the kij loop placing the pragma omp for on the k loop with k, l, j as the private data clause result in about 6.5 times the performance and no data difference errors. Placing the pragma command on the i loop results between 4 and 7 times the performance and no data difference errors. Placing the pragma command at the j loop result in incorrect results and about 1/5 in performance.

For the ijk loop placing the pragma omp on the outer l loop with l,j and k in the private data clause results between 4 and 6 times the performance and no data difference errors. Placing the pragma omp on the j loop results in between 4 and 6 times the performance and no data difference errors. Placing the pragma on the k loop result in another a small drop in performance and data validity error.

- b. My best performance for the ijk loop was 6.06 GFLOPS. For the kij loop 10 GFLOPS. In the ijk version I added the schedule(dynamic) clause to the outermost for loop. In the KIJ loop I added the schedule(auto) caluse to the l loop as I found that it gave me the best performance. I also tried unrolling the innermost loop but that result in a program that got stuck. The best I was able to do was 13 times base performance.
4. My best merge sort run was 130 Mega-Elements/sec. Unfortunately that run had some data error in it that I was not able to solve in the time limit. In testing I can properly sort the latter half of the array but I can't get the first half sorted.