Homework 7 : Exercise 4 Vectorsum

Implementation 1:

```
#pragma omp parallel for
   for(int iloop = 0; iloop<nloops; iloop++){
     for(int i =0; i<vectorsize; i++){
       outvec[i] += invec[i]*loopcoeff[iloop];
     }
   }
```

Only outer (iloop) is parallelized and each thread gets a chunk of iloop to work on while but the inner loop is still executed sequentially within each thread.

Vector Size: 1,000,000 : Optimization level 2
Sequential t = 0.28970 sec
Threads 4 t = 0.07256 sec

Vector Size: 10,000,000 : Optimization level 2
Sequential t= 4.61640 sec
Threads 4 t= 1.34130 sec

Vector Size: 10,000,000 : Optimization level 3
Sequential t= 2.57050 sec
Threads 4 t= 1.58670 sec

Implementation 2:

```
#pragma omp parallel for collapse(2)
for(int iloop = 0; iloop<nloops; iloop++){
 for(int i=0; i<vectorsize; i++){
   outvec[i] += invec[i] * loopcoeff[iloop];
 }
}
```

Two nested loops are collapsed into a single 2D iteration space. In this situation OpenMP distributes the combined iteration space of loops * vectorsize. This could theoretically improve load balancing and performance.

Vector Size: 1,000,000 : Optimization level 2

Sequential t = 0.28979 sec
Threads 4 t = 0.15675 sec

Vector Size: 10,000,000 : Optimization level 2
Sequential t = 4.58867 sec
Threads 4 t = 0.20746 sec

Vector Size: 10,000,000 :
Sequential t= 2.31115 sec
Threads 4 t= 0.19098 sec

Conclusion

Both implementations yield similar sequential timings, but their parallel performance differs notably. In implementation 1, only the outer loop is parallelized, which works efficiently for smaller vector sizes, as seen with 1,000,000 elements. However, for larger vector sizes (10,000,000) and with higher optimization levels, collapsing both loops in implementation 2 dramatically improves parallel performance. This indicates that the overhead from collapsing can hurt performance on smaller performance. This indicates that while the overhead from collapsing can hurt performance on smaller problems, it provides better load balancing and scalability for larger workloads.