Monique Nguyen
CSC656
04/22/25

SECONDS

Problem Size	Blas	Basic	Vectorized	omp-1	omp-4	omp-16	omp-64
1024	0.00049	0.00091	0.00032	0.00094	0.00107	0.00315	0.01025
2048	0.00057	0.00358	0.00094	0.00359	0.00188	0.00224	0.00413
4096	0.00389	0.01461	0.00417	0.01467	0.00430	0.00432	0.00713
8192	0.01781	0.05827	0.01683	0.05829	0.01588	0.01495	0.01690
16384	0.07449	0.23338	0.06727	0.23317	0.06043	0.05745	0.06152

MFLOPS
(2*N^2) / seconds * 10^6

Problem Size	Blas	Basic	Vectorized	omp-1	omp-4	omp-16	omp-64
1024	4279.90	2304.56	6553.6	2331.01	1959.95	665.76	204.60
2048	14716.85	2343.18	8924.05	2336.65	4462.02	3744.91	2031.13
4096	8625.82	2296.67	8046.62	2287.28	7803.35	7767.22	4706.09
8192	7536.08	2303.37	7974.90	2302.58	8452.99	8977.77	7941.87
16384	7207.28	2300.41	7980.83	2302.48	8884.17	9345.01	8726.77

MEMORY BANDWIDTH
Bytes accessed = 8N^2 +16N
% =(bytes/t)/(204.8*10^9) x100

Problem Size	Blas	Basic	Vectorized	omp-1	omp-4	omp-16	omp-64
1024	8.38%	4.51%	12.83%	4.37%	3.84%	1.30%	0.40%
2048	28.77%	4.58%	17.45%	4.57%	8.72%	7.32%	3.97%

4096	16.86%	4.49%	15.72%	4.47%	15.25%	15.18%	9.20%
8192	14.72%	4.50%	15.58%	4.50%	16.51%	17.54%	15.52%
16384	14.08%	4.49%	15.59%	4.50%	17.35%	18.25%	17.05%

In this assignment the way I ran it was option 2 : scripted

Analysis Questions

1. For my code the one that had the best MFLOPS was my vectorized implementation. For my basic implementation my MFLOPS was 2300.41 while my vectorized implementations had 7980.83. Meaning vectorized is at least 2x faster than basic. This will go the same for our memory system utilization. For our vectorized implementation its at a 15.59% compared to basic which was at only a 4.49%
2. When it comes to comparing MFLOPS between the OpenMP 4-way and basic, the OpenMP beats the basic. OpenMP 4 way has 8884.17 MFLOPS compared to basic at only 2300.41 meaning OpenMP is at least 2x faster. For system utilization the same comparison results. OpenMP memory utilization is at a 17.35% while basic is only at a 4.49%
3. 1 thread ->0.23317
   4 thread ->0.06043
   16 thread ->0.05745
   64 thread ->0.06152

   Speedup
   1-4 threads -> 0.23317/0.06043 = 3.86
   1-16 threads -> 0.23317/0.05745 = 4.06
   1-64 threads -> 0.23317/0.06152 = 3.79