Nikhil Rajendra Danapgol HPC Assignment 2 2019BTECCS00036

Program for Vector to Vector Addition

```
    vectorAdditionPara.cpp > 分 main()

      #include <omp.h>
      #include <pthread.h>
      #include <stdio.h>
      int main()
          int N = 1000;
          int mat1[1000];
          for (int i = 0; i < N; i++)
              mat1[i] = 10;
 11
          int mat2[1000];
 12
          for (int i = 0; i < N; i++)
              mat2[i] = 20;
          int ans[1000] = \{0\};
          double stime = omp get wtime();
      #pragma omp parallel for reduction(+ \
 19
                                         : ans)
          for (int i = 0; i < N; i++)
              ans[i] = mat1[i] + mat2[i];
              printf("Index: %d Thread: %d\n", i, omp get thread num());
```

```
for (int i = 0; i < N; i++)
{
    printf("%d ", ans[i]);
}

double etime = omp_get_wtime();
double time = etime - stime;
printf("\nTime taken is %f\n", time);
printf("\n");
return 0;
}</pre>
```

Output:

```
Index: 227 Thread:
Index: 228 Thread:
Index: 229 Thread:
Index: 230 Thread:
Index: 231 Thread:
Index: 232 Thread:
Index: 233 Thread:
Index: 234 Thread: Index: 235 Thread:
Index: 236 Thread:
Index: 237 Thread:
Index: 238 Thread:
Index: 239 Thread:
Index: 240 Thread:
Index: 241 Thread:
Index: 242 Thread:
PS C:\HPCLAB\Assignment2>
```

Sequential Algorithm:

```
vectorAdditionSeq.cpp > ...
   #include <omp.h>
   #include <pthread.h>
   #include <stdio.h>
   int main()
        int N = 1000;
        int mat1[1000];
        for (int i = 0; i < N; i++)
            mat1[i] = 10;
        int mat2[1000];
        for (int i = 0; i < N; i++)
            mat2[i] = 20;
        int ans[1000] = \{0\};
       double stime = omp_get_wtime();
        for (int i = 0; i < N; i++)
            ans[i] = mat1[i] + mat2[i];
            printf("Index: %d Thread: %d\n", i, omp get thread num());
        for (int i = 0; i < N; i++)
            printf("%d ", ans[i]);
       double etime = omp_get_wtime();
       double time = etime - stime;
        printf("Time taken is %f\n", time);
        return 0;
```

Output:

Index: 995 Thread: 0 Index: 996 Thread: 0 Index: 997 Thread: 0 Index: 998 Thread: 0

Time taken is 0.259000
PS C:\HPCLAB\Assignment2>■

Program for Vector-Scalar Multiplication:

Sequential:

```
• vectorScalarMultiplicationSeq.cpp > • main()
      #include <omp.h>
      #include <pthread.h>
      #include <stdio.h>
      int main()
          int N = 500;
          int mat1[501];
          for (int i = 0; i < N; i++)
              mat1[i] = 10;
 11
          int S = 6;
 12
          double itime;
          itime = omp_get_wtime();
          for (int i = 0; i < N; i++)
 17
              mat1[i] *= S;
              printf("Index: %d Thread: %d\n", i, omp get thread num());
          for (int i = 0; i < N; i++)
              printf("%d ", mat1[i]);
 24
          double ftime = omp_get_wtime();
          double exec time = ftime - itime;
          printf("\nTime taken is %f\n", exec_time);
          printf("\n");
```

Output:

Index: 494 Inread: 0					
Index: 495 Thread: 0					
Index: 496 Thread: 0					
Index: 497 Thread: 0					
Index: 498 Thread: 0					
Index: 499 Thread: 0					
60 60 60 60 60 60 60 60 60 60 60 60 60 6	60 60 60	60 60 60 6	0 60 60	60 60 60	9 60 60 6
9 60 60 60 60 60 60 60 60 60 60 60 60 60	9 60 60 6	50 60 60 60	60 60 6	60 60	60 60 60
60 60 60 60 60 60 60 60 60 60 60 60 60 6	60 60 60	60 60 60	60 60 60	60 60 6	50 60 60
60 60 60 60 60 60 60 60 60 60 60 60 60 6	60 60 60	60 60 60 6	0 60 60	60 60 60	9 60 60 6
0 60 60 60 60 60 60 60 60 60 60 60 60 60	0 60 60 6	50 60 60 60	60 60 6	60 60 60	60 60 60
60 60 60 60 60 60 60 60 60 60 60 60 60 6	60 60 60	60 60 60	60 60 60	60 60 6	50 60 60
60 60 60 60 60 60 60 60 60 60 60 60 60 6	60 60 60	60 60 60 6	0 60 60	60 60 60	9 60 60 6
0 60 60 60 60 60 60 60 60 60 60 60 60 60	0 60 60 6	50 60 60 60	60 60 6	0 60 60	60 60 60
60 60 60 60 60 60 60 60 60 60 60 60 60 6	60 60 60	60 60 60	60 60 60	60 60 6	60 60 60
60 60 60 60 60 60 60 60 60 60 60 60 60 6	60 60 60	60 60 60 6	0 60 60	60 60 60	9 60 60 6
0 60 60 60 60 60					
Time taken is 0.111000					
PS C:\HPCLAB\Assignment2>					
	Ln 24, Col 6	Spaces: 4	UTF-8	CRLF C+	-+ 🏟 Go Li

Parallel:

```
vectorScalarMultiplicationPara.cpp > 🗘 main()
     #include <omp.h>
     #include <pthread.h>
     #include <stdio.h>
     int main()
         int N = 500;
         int mat1[501];
         for (int i = 0; i < N; i++)
             mat1[i] = 20;
11
         int S = 6;
12
13
         double itime;
14
         itime = omp get wtime();
     #pragma omp parallel for
15
         for (int i = 0; i < N; i++)
16
17
18
             mat1[i] *= S;
19
             printf("Index: %d Thread: %d\n", i, omp_get_thread_num());
20
21
22
         for (int i = 0; i < N; i++)
23
             printf("%d ", mat1[i]);
24
25
26
27
         double ftime = omp get wtime();
28
         double exec time = ftime - itime;
29
         printf("\nTime taken is %f\n", exec_time);
         printf("\n");
30
```

Output:

```
PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE

Index: 494 Thread: 7
Index: 495 Thread: 7
Index: 495 Thread: 7
Index: 496 Thread: 7
Index: 497 Thread: 7
Index: 497 Thread: 7
Index: 498 Thread: 7
Index: 499 Thread: 7
Ind
```

Github Link:

https://github.com/nd22052000/HPC