Assignment 4: OpenMP - loops

The purpose of this assignment is for you to learn more about

- the parallel loop construct of OpenMP,
- how loop scheduling works in OpenMP,
- how easy (or not so easy) is writting parallel codes with OpenMP loop construct.

As usual all time measurements are to be performed on the cluster.

Activate OpenMP in GCC by passing `-fopenmp` to the compiler and linker. (Note that if you omit this parameter, the code will probably still compile. But its execution will be sequential.)

You can control the number of threads in OpenMP in two ways: the environment variable `OMP_NUM_THREADS` or by calling the `omp_set_num_threads` function. Use the function call.

Loop scheduling in OpenMP can be controlled either by specifying `schedule(runtime)` in the parallel for construct and specifying a `OMP_SCHEDULE` environment variable, or by using the `omp_set_schedule` function. Use the function call.

The first two problems are here to kickstart your OpenMP programming and why people love OpenMP (and only use pthreads when necessary). The latter one are more complex.

1 Reduce

Question: Write a program that will generate a random array of integers. Then it will compute the sum of the array in parallel using the OpenMP for loop construct. Measure the time computing the sum takes. Write the program so that you can control the length of the array, the number of thread and the scheduling policy. Output the value of the reduction to stdout and the time taken to compute it to stderr.
Question: Plot a speedup chart for different numbers of thread (1, 2, 4, 8, 16), for arrays of sizes 10^8, with different scheduling policies (static, dynamic,1, dynamic,1000, dynamic,100000) using the bench script. Does the plot make sense? Why?

2 Numerical Integration

Question: Take your sequential code to compute numerical integration from the previous assignment. Modify it to do the numerical integration in parallel with the OpenMP loop construct. Output the value of the integration to stdout and the time taken to compute it to stderr.
Question: Plot a speedup chart for different numbers of thread (1, 2, 4, 8, 16), for different scheduling policies (dynamic,1 and dynamic,1000), for different number of points (N=10^3 and N=10^9), and intensity (10, and 1000) using the bench script. Does the plot make sense? Why?

3 Prefix Sum

Here is a sequential Prefix Sum

```
//arr of length n. pr of length n+1
void prefixsum (int* arr, int n, int* pr) {
  pr[0] = 0;
  for (int i=1; i<=n; ++i)
    pr[i] = pr[i-1] + arr[i-1];
}
```

Question: Implement a parallel function using OpenMP parallel loop constructs to compute the prefix sum of an array. Pick the scheduling policy you feel is appropriate. Output the time it took on stderr.
Question: Plot a speedup chart for different numbers of thread (1, 2, 4, 8, 16), for arrays of sizes 10^9 using the bench script. Does the plot make sense? Why?

4 Merge Sort

Question: Implement a parallel function using OpenMP parallel loop constructs to perform merge sort on an array of integer. Output the time it took on stderr.
Question: Plot a speedup chart for different numbers of thread (1, 2, 4, 8, 16), for arrays of sizes 10^9 using the bench script. Does the plot make sense? Why?