Parallel For Loops: Prefix Sum

Question: Before starting, run all sequential codes on `Centaurus` using `make bench`.

1 Prefix Sum

Here is a sequential Prefix Sum:

```
void prefixsum (int* arr, int n, int* pr) {
  pr[0] = 0;

  for (int i=0; i<n; ++i) {
    pr[i+1] = prefix[i] + arr[i];
  }
}
```

Question: Implement a parallel function using parallel loop constructs to compute the prefix sum of an array. Output the time it took on `stderr`. Use the template provided in `prefixsum/prefixsum.cpp`. Note that the data is generated by function `generatePrefixSumData()` and the correctness of the result is checked by function `checkPrefixSumResult`. Remember to set thread count and granularity using the `setNbThread()` and `setGranularity()` functions provided in the `omploop.hpp` file. Output the time it took on `stderr`.

Question: Run the code on centaurus, in the `prefixsum/` directory, using `make bench`. And then plot the results using `make plot`. Does the plot make sense? Why?