Loop-carried dependencies

November 22, 2012

The file `loopdependencies.c` contains a main program that allocates two arrays `a` and `b` of size `n` and the performs the following operations on their coefficients:

```
for(i=0; i<n; i++){
  b[i] = a[i]+1;
  if(i != 0)
    a[i] = b[i]-b[i-1];
}
```

The objective of this exercise is to parallelize the code above using OpenMP.

1 Package content

The directory `LoopDependencies` contains a single file named `loopdependencies.c`. This file contains a main program with the code described above and an **incorrect** parallel version of it (it is explained in the next section how to use it). The code can be compiled with the `make` command: just type `make` inside the `LoopDependencies` directory. This will generate a `main` executable file that can be executed like this

```
$ ./main
```

This program will execute both the sequential version of the code reported above and the parallel version and print a message saying whether the parallel version computed a correct result or not. At the beginning the `loopdependencies.c` file contains an incorrect parallel version; your objective is to modify it in order to compute a correct result.

2 Assignment

- ✎ in the `responses.txt` file, explain why the following parallel version (contained in the initial version of the `loopdependencies.c` file) is incorrect:

  ```
  #pragma omp parallel for
  for(i=0; i<n; i++){
    b[i] = a[i]+1;
    if(i != 0)
      a[i] = b[i]-b[i-1];
  }
  ```

- ⌨ propose a correct OpenMP parallel version; more precisely, modify (or replace) the incorrect parallel version in the `loopdependencies.c` file in order to compute the correct result. Compile the code and run it to verify that the modified version actually computes the correct result.