

Supercomputing in Plain English

Exercise #4: Loop Carried Dependencies

In this exercise, we'll use the same conventions and commands as in Exercises #1, #2 and #3. You should refer back to the Exercise #1, #2 and #3 descriptions for details on various Unix commands.

In the exercise, you'll benchmark various array operations, some of which have loop carried dependencies, using various compilers and levels of compiler optimization.

Specifically, you'll benchmark using the following compilers:

- the GNU Fortran compiler, `gfortran`, for various optimization levels;
- the Intel Fortran compiler, `ifort`, for various optimization levels;
- the Portland Group Fortran compiler, `pgf90`, for various optimization levels.

Here are the steps for this exercise:

1. Log in to the Linux cluster supercomputer (`sooner.oscer.ou.edu`).

2. Copy the `LoopCarriedDependencies` directory:

```
% cp -r ~hneeman/SIPE2011_exercises/LoopCarriedDependencies/ ~/SIPE2011_exercises/
```

3. Choose which language you want to use (must be C), and `cd` into the appropriate directory:

```
% cd ~/SIPE2011_exercises/LoopCarriedDependencies/C/
```

NOTE : This exercise doesn't currently have a Fortran90 version, so C is the only option.

4. Edit the batch script `loop_carried_dependencies.bsub` so that it contains your username and your e-mail address.

5. Compile, using the *shell script* named `make_cmd` (a shell script is a file containing a sequence of Unix commands), which in turn invokes the `make` command:

```
% make_cmd
```

If that doesn't work, try this:

```
% ./make_cmd
```

6. Submit the batch job:

```
% bsub < loop_carried_dependencies.bsub
```

7. Once the batch job completes, examine the various output files to see the timings for your runs with executables created by the various compilers under the various levels of optimization.

8. Use your favorite graphing program (for example, Microsoft Excel) to create graphs of your various runs, so that you can compare the various methods visually.