Driver: Julia Sales - Senior, Computer Science
Navigator: Anthony Overton - Senior, TIM
Group: 44-3
10/17/2019

Homework 2- Fortran Matrix Multiplication

HW2	Julia Sales	Anthony Overton
Coding	90% (main coding)	10% (helped debug)
Results	50% (ran tests)	50% (ran tests)
Report	20% (wrote a section)	80% (main writer)
Overall	Driver	Navigator

Explanation of Flag Options[1]:

- optimization with `-O3 -fdefault-real-8 -fdefault-double-8`

  -O3 Highest level of optimization. Higher levels usually produce faster code but take longer to compile.

  fdefault-real-8 Set the default real type to an 8 byte wide type. This option also affects the kind of non-double real constants like 1.0. This option promotes the default width of DOUBLE PRECISION and double real constants like 1.d0 to 16 bytes if possible.

  fdefault-double-8 Set the DOUBLE PRECISION type and double real constants like 1.d0 to an 8 byte wide type. Do nothing if this is already the default. This option prevents -fdefault-real-8, -fdefault-real-10, and -fdefault-real-16, from promoting DOUBLE PRECISION and double real constants like 1.d0 to 16 bytes.

- optimization with `-O0 -fdefault-real-8 -fdefault-double-8`

  -O0 Default optimization (none).

- debug flags with `-g3 -fdefault-real-8 -fdefault-double-8 -Wall -Wextra -Wimplicit-interface -fPIC -fmax-errors=1 -g -fcheck=all -fbacktrace`

  -g3 Generates extra debugging information.

[1] https://gcc.gnu.org/onlinedocs/gfortran/Option-Summary.html

-Wall “Warn about all” generates warnings about common sources of bugs.

-Wextra warns about subroutine arguments that are never used, which is almost always a bug.

-Wimplicit-interface request warnings on implicit declarations. Warn if a procedure is called without an explicit interface. Note this only checks that an explicit interface is present. It does not check that the declared interfaces are consistent across program units.

-fPIC[2] Position Independent Code means that the generated machine code is not dependent on being located at a specific address in order to work. If compiled with -fPIC, it's suitable for inclusion in a library.

-fmax-errors=1 Limits the maximum number of error messages to n, at which point GNU Fortran bails out rather than attempting to continue processing the source code. If n is 0, there is no limit on the number of error messages produced.

-fcheck=all Enable all run-time test of -fcheck.

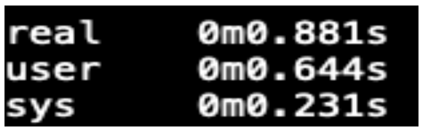
-fbacktrace Specify that, when a runtime error is encountered or a deadly signal is emitted (segmentation fault, illegal instruction, bus error or floating-point exception), the Fortran runtime library should output a backtrace of the error. This option only has influence for compilation of the Fortran main program.

__**Number of Nonzero Entries in the result of C1-C2:**__ n = 4096: 16773120 & n = 4097: 16781312

__**Analysis of Runtimes under Different Flags:**__

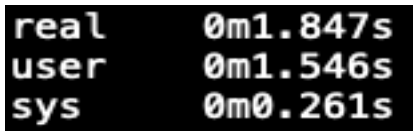
For `array_col_hw2.f90`:

n = 4096: Maximum Optimization



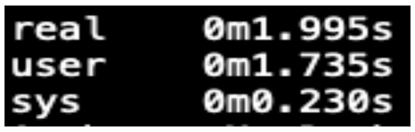
```
real    0m0.881s
user    0m0.644s
sys     0m0.231s
```

n = 4096: Default Optimization (none)



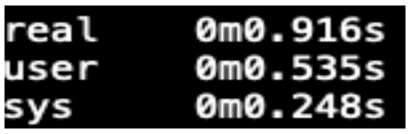
```
real    0m1.847s
user    0m1.546s
sys     0m0.261s
```

n = 4096: Debug Flags



```
real    0m1.995s
user    0m1.735s
sys     0m0.230s
```

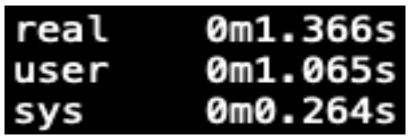
n = 4097: -O3



```
real    0m0.916s
user    0m0.535s
sys     0m0.248s
```

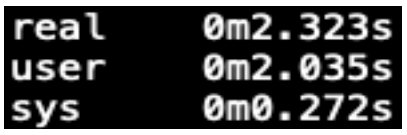
[2] https://stackoverflow.com/questions/5311515/gcc-fpic-option

n = 4097: -O0



```
real	0m1.366s
user	0m1.065s
sys	0m0.264s
```

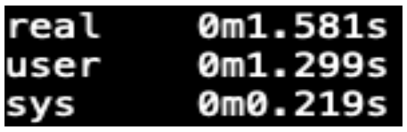
n = 4097: Debug Flags



```
real	0m2.323s
user	0m2.035s
sys	0m0.272s
```

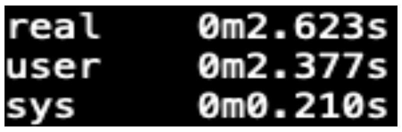
For `array_row_hw2.f90`:

n = 4096: Maximum Optimization



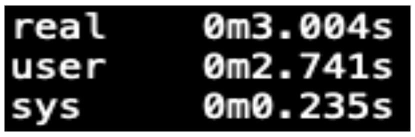
```
real	0m1.581s
user	0m1.299s
sys	0m0.219s
```

n = 4096: Default Optimization



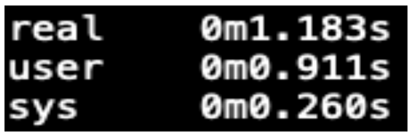
```
real	0m2.623s
user	0m2.377s
sys	0m0.210s
```

n = 4096: Debug Flags



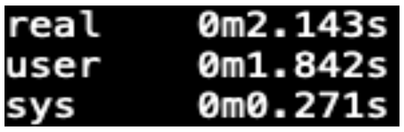
```
real	0m3.004s
user	0m2.741s
sys	0m0.235s
```

n = 4097: -O3



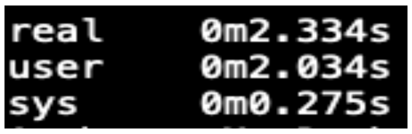
```
real	0m1.183s
user	0m0.911s
sys	0m0.260s
```

N = 4097: -O0



```
real	0m2.143s
user	0m1.842s
sys	0m0.271s
```

N = 4097: Debug Flags



```
real	0m2.334s
user	0m2.034s
sys	0m0.275s
```

Conclusion:

To begin, the program generally ran faster with the -O3 maximum optimization flag, than it did with no optimization (-O0).

Fortran handles multi-dimensional arrays in column-major convention. This means that the *columns* (of a matrix for example) are stored contiguously in memory. The significance of contiguity stored data is that the process can locate this data very fast. The programmer can use this to his or her advantage when writing and algorithm. In our case, the column order convention generally ran faster due to the column elements being continuously stored in memory.

4096 vs 4097: The non-power of 2 number 4097 should run faster (CPU time), because of how the CPU cache handles data. The power of 2 number evenly fits in the cache, while another number may have leftover data that will aid the process in the next cycle. It is against our expectation that the larger number would be faster.