SALIH KILICLI – STAT 624 HOMEWORK 2

—— SETUP ——

```
module load intel/2019
```

I have written a shell file that runs a couple for loops and compares the Intel and GNU libraries and their options in terms of performance. The file is called hw2.sh, and the code is given below.

Finally, I made the shell file executable and then run it on my Terra account.

```
chmod u+x hw2.sh

./hw2.sh
```

—— HW2.SH FILE ——

```
# Compiler Optimization Comparison - Intel vs GNU with different options

# Runs a for loop for different options for each compiler and times the process

# O0 (No optimization), O1 (Some for speed), O2 (All for speed), O3 (Aggressive), -fast (or -Ofast for GNU)

IC=("-O0" "-O1" "-O2" "-O3" "-fast")

GNU=("-O0" "-O1" "-O2" "-O3" "-Ofast")

echo "_____"

echo          "Comparing Intel Fortran vs GNU Fortran w/ different optimizers"

echo "_____"

for i in {0..4}

        do

                ifort ${IC[$i]} -o brem brem.f gammln.f

                echo "Time for" ${IC[$i]} "optimization of Intel compiler is"

                time ./brem < thik.inp $> /dev/null

                gfortran ${GNU[$i]} -o brem brem.f gammln.f

                echo "Time for" ${GNU[$i]} "optimization of GNU compiler is"

                time ./brem < thik.inp $> /dev/null

done
```

```
echo "_____"

echo          "Comparing Intel compiler options along with -O2 (default) optimizer"

echo "_____"

ICO=("-xHost" "-ipo" "-no-prec-div")

for j in {0..2}

        do

                ifort ${ICO[$j]} -o brem brem.f gammln.f                    #Intel compiler

                echo "Time for Intel compiler option" ${ICO[$j]} "along w/ -O2 is"

                time ./brem < thik.inp $> /dev/null

done

echo "_____"

echo "Comparing GNU optimizer along with GNU vs Intel math library

echo "_____"

                gfortran -Ofast -o brem brem.f gammln.f                     #GNU compiler

                echo "Time for -Ofast optimization of GNU w/ GNU math library is"

                time ./brem < thik.inp $> /dev/null

                gfortran -Ofast -o brem brem.f gammln.f -limf     #GNU compiler with Intel math library

                echo "Time for -Ofast optimization of GNU w/ Intel math library is"

                time ./brem < thik.inp $> /dev/null

echo "_____"
```

—— FINAL REMARKS ——

Running hw2.sh executable file in Terra and collecting time information, I have created tables below in order to compare required optimizations and their options.

Table below compares different optimization methods for Intel and GNU compilers and compare them within the same optimization method.

SALIH KILICLI – STAT 624 HOMEWORK 2

Compiler Type	Intel	GNU
-O0	real 1m27.730s user 1m27.727s sys 0m0.005s	real 2m34.567s user 2m34.565s sys 0m0.007s
-O1	real 0m48.791s user 0m48.785s sys 0m0.007s	real 2m20.027s user 2m20.023s sys 0m0.008s
-O2	real 0m47.627s user 0m47.619s sys 0m0.009s	real 2m7.949s user 2m7.942s sys 0m0.009s
-O3	real 0m47.436s user 0m47.426s sys 0m0.011s	real 2m5.223s user 2m5.216s sys 0m0.011s
-fast (-Ofast GNU)	real 0m15.885s user 0m15.882s sys 0m0.004s	real 2m7.103s user 2m7.054s sys 0m0.034s

It is evident from the table that for Intel compiler, fastest method is given by -fast option which is followed by -O2 and -O3 being really close to each other. For GNU, apparently, -O3 method works the best; however, the speed for -O3, -O2, and -Ofast options are pretty similar. Also, it is apparent that Intel compiler works much better than GNU compiler when it comes to speed. Even the fastest method of GNU is much slower that the slowest method of Intel.

Table below compares different options for Intel compiler using the default (-O2) optimization method.

Options	-xHost	-ipo	-noprec-div
Intel	real 0m47.097s user 0m47.092s sys 0m0.005s	real 0m16.937s user 0m16.932s sys 0m0.006s	real 0m47.161s user 0m47.156s sys 0m0.006s

Clearly, -ipo option gives the biggest performance boost in terms of speed, and it almost works 3x times faster than the -O2 optimizer without the given option. The other two option yields pretty similar results.

The table below illustrates the difference of speed performance of GNU library's -Ofast optimizer along with default and Intel math libraries.

Library	GNU Math Library	Intel Math Library
GNU -Ofast optimizer	real 2m6.450s user 2m6.444s sys 0m0.009s	real 0m52.658s user 0m52.651s sys 0m0.006s

Again. It is clear that GNU compiler works more than 2x times faster when it is used with Intel math library (even when compared to fastest optimization method), which means the mathematical library that is used is more important than the compiler used in the code when it comes to speed.