ECS230 Applied Numerical Linear Algebra   Fall 2023   2023-11-03

Homework 3 due Tue 2023-11-14 at 23:59

Create a pdf file containing your report. Upload a zip file hw3.zip containing your programs and your pdf report.

1) Matrix multiplication
Write a program that implements matrix multiplication $C=AB$ where A, B and C are $(n \times n)$ real matrices represented as double-precision numbers. The value of n must be read as a command line argument by the program. The program must use dynamic memory allocation for all matrices. The program must initialize the matrices A and B with non-zero arbitrary values of your choice before starting the calculation of the product. Determine the total number of floating-point operations performed in the product. Use the **`gtod()`** function (provided in Homework2) to measure the time taken to compute the product. Use repeated runs to ensure that timings are accurate. Compute the floating point performance (in GFlop/s) using the total number of operations and the time elapsed. The program must print the following information on output on two separate lines:
- the elapsed time (in seconds).
- the computed floating-point performance in GFlop/s, printed as a floating-point number.

a) Provide results for the following values of n: 100 200 500 1000 2000.
b) Comment on your results and compare with the expected floating-point performance of the processor. Note that processors are different depending on which CSIF PC you are using (pc01-pc15 have a Core i7-9700 CPU 3.0 GHz and pc16-pc45 have a Core i7-10700 2.9 GHz). The peak performance of one core depends on the width of SIMD registers, clock frequency, hyperthreading, and the availability of a fused multiply-add FMA instruction. The peak performance (all 8 cores) is 384 GFlop/s for the Core i7-9700 CPU and 371.2 GFlop/s for the Core i7-10700 CPU. Compute the fraction of the peak performance of a single core achieved in your runs.

2) Write a modified version of the above program implementing all 6 possible loop orderings in a matrix multiplication. Run the program and compare the performance for each loop ordering. Compare results obtained when compiling with gcc without options, and with the **`-O0`**, **`-O2`** and **`-O3`** optimization options.

3) Write a program implementing matrix multiplication **`using`** the **`dgemm`** function of the BLAS library (http://www.netlib.org/blas). The library is installed on CSIF computers. Use the netlib.org web page to get details about the arguments of the **`dgemm`** function. Note that **`dgemm`** is a Fortran function and all parameters from a C program should be passed through pointers. Use the **`dotblas.c`** example program (provided on Canvas) as an example of how to call BLAS functions. Use the additional "-lblas" argument during compilation to link to the BLAS library. Measure timings for the above problem sizes and discuss your results.
Note: all programs in this homework assignment should use only one core of the processor. However, the BLAS library is multithreaded. Use the command
"export OMP_NUM_THREADS=1" while running your program to make sure that you measure the performance of one core only. In your analysis, consider how the "turbo boost" feature of Intel processors may affect the results.