High Performance Computing : Home Work 3

SACHIN SATISH BHARADWAJ
N16360220 (ssb638)
https://github.com/SachinSBharadwaj/hw3.git

April 5th, 2020

Question 2
Taylor Series and Vectorisation

(1) EXTRA CREDIT:

To implement the function evaluation **outside** the region $x \in [-\pi/4, \pi/4] \Rightarrow x \in [\pi/4, 7\pi/4]$, the following was done (**NOTE**: Another code called **fast-sin-extra.cpp** contains the code for this extra credit question. Thus, after doing *make*, run the **./fast-sin-extra** executable for the results of this question):

1. The (a) symmetries of sine (b) Relation between sine and cosine was used.
2. Since evaluating $\sin(x)$ for $x \in [-\pi/4, \pi/4]$ gives higher accuracy when sine is expanded using Taylor series around $x = 0$. So for the region outside of this, we need to use symmetry arguments and sine-cosine relation to bring the angle into this favourable region and then evaluate.
3. For this first we observe the following facts:
   (a) Sine function is positive in 1st and 2nd quadrants while it is negative in the 3rd and 4th quadrants.
   (b) $\sin(\pi/2 - \theta) = \cos(\theta)$
   (c) $\sin^2(\theta) + \cos^2(\theta) = 1$
4. Now, first we generate random numbers or angles for $x \in [\pi/4, 7\pi/4]$.
5. Then we convert each of these to $x \in [-\pi/4, \pi/4]$. Along with this considering the above observations in point 3, the final evaluation would be as follows:

$$\sin(x) = \begin{cases} \cos(\pi/2 - x) = \sqrt{1 - \sin^2(\pi/2 - x)} & x \in (\pi/4, \pi/2] \\ \cos(x - \pi/2) = \sqrt{1 - \sin^2(x - \pi/2)} & x \in (\pi/2, \pi] \\ -\cos(3\pi/2 - x) = -\sqrt{1 - \sin^2(3\pi/2 - x)} & x \in (\pi, 3\pi/2] \\ -\cos(x - 3\pi/2) = -\sqrt{1 - \sin^2(x - 3\pi/2)} & x \in (3\pi/2, 7\pi/4] \end{cases}$$

6. In the above mapping, we also create another array that holds the sign of the quadrant information and is correspondingly multiplied in the subroutines. Along with this two more terms of Taylor series is added.
7. The same evaluation is done for both sin4_taylor() and AVX vectorised functions.
8. This procedure yielded an accuracy upto 4th decimal, which can be further improved.

Question 3
Parallel Scan in OpenMP

(1) MACHINE SPECS: First, let us look at the processor/machine details on which **all the** codes were run on. A brief summary of the processor information would look something like the following:

1. **CPU** :
   (a) Machine: Lenovo ThinkStation P330 (2nd Gen)
   (b) Processor Company: Genuine Intel
   (c) CPU Family: 6
   (d) Model Name: Intel(R) Core(TM) i9-9900 CPU @ 3.10GHz
   (e) Stepping: 1
   (f) Siblings / CPU cores : 8 / 16 processors
   (g) Number of Threads: 16
   (h) CPU MHz = 1600.006 MHz
   (i) CPU Max Frequency: 3.1 GHz
   (j) Max Turbo Speed : 5 GHz
   (k) Max FLOPS : ≈ 300 GFlops/s
2. **MEMORY** :
   (a) Mem Total (Slow Access) : 1 TB
   (b) RAM: 32 GB
   (c) L1 Cache : 512 KB
   (d) Cache Size: 16384 KB
   (e) Rated Memory Speed : 2.66 MHz
   (f) g++/gcc compiler version : gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1 18.04)

(2) PERFORMANCE REPORT:

1. THREADS = 2
   (a) sequential-scan = 0.205109s
   (b) parallel-scan = 0.199846s
   (c) error = 0
2. THREADS = 4
   (a) sequential-scan = 0.202154s
   (b) parallel-scan = 0.133884s
   (c) error = 0
3. THREADS = 8
   (a) sequential-scan = 0.211572s
   (b) parallel-scan = 0.107716s
   (c) error = 0
4. THREADS = 10
   (a) sequential-scan = 0.204219s
   (b) parallel-scan = 0.111579s
   (c) error = 0
5. THREADS = 16
   (a) sequential-scan = 0.217273s
   (b) parallel-scan = 0.134258s
   (c) error = 0