High Performance Scientific Computing
Homework Assignment 1

Naveen Himthani (120010001)

Spring 2015-16

1 Question 1

Using Amadahl's law,

$$S(N) = \frac{1}{(1-P) + \frac{P}{N}} \tag{1}$$

where for the given problem $(1 - P) = 20\%$

1. Maximum speed-up achievable will be when $N = \infty$, therefore

$$S = \frac{1}{0.2 + 0} \tag{2}$$

$$\mathbf{S = 5} \tag{3}$$

2. For a desirable speed-up of $S = 50$, the maximum serial portion of the algorithm can again be calculated when $N = \infty$

$$50 = \frac{1}{(1-P) + 0} \tag{4}$$

$$1 - P = 0.02 \tag{5}$$

therefore, the **serial portion of the algorithm is** 2%

2 Question 2

Assumption $\Rightarrow$ sizeof(int) = 4 bytes Total size of the data in main memory = $\frac{1024 \times 1024 \times 4}{1024}$ = 4096 Kb

2.1 CPU cycles in C program

Since data is stored in the row-major format in C and the algorithm is written such that for each row a column is scanned, the data is read in a sequential manner without having to fetch large chunks of data for reading the next element as the algorithm requires.

(a) Cost of fetching 4096 Kb from main memory to cache = $4096 \times 150 = 614400$ CPU Cycles

(b) Cost of fetching 1 Kb from Cache to CPU = $\frac{1024 \times 1}{8} = 128$ CPU Cycles

(c) Therefore total cost of the algorithm = $614400 + 128 \times 4096 = 1138688$ CPU Cycles

2.2 CPU cycles in FORTRAN program

Since data is stored in the column-major format in FORTRAN and the algorithm is written such that for each row a column is scanned, the data is not read in a sequential manner i.e. it has to unnecessarily fetch large chunks of data for reading next element.

(a) For accessing each element from the array in the order required by the algorithm, the cache will have to access 1 Kb data (256 integers) from main memory at a time which will incur 150 CPU cycles. Out of those 256 integers, the CPU only needs the first element every time for which it will access the first 2 elements (8 bytes) incurring 1 CPU cycle. In short, for getting 1 integer from the main memory to CPU costs $150 + 1 = 151$ CPU cycles.

(b) This has to be done for all 1024×1024 elements of the array incurring $1024 \times 1024 \times 151 = 158334976$ CPU Cycles.