HW8

1. Given, two applications runs on this 8-core processor, but the resource requirements are not equal. The first application needs 80% of the resources, and the other only 20% of the resources.
   And Speedup = 1 / ((1-f)+f/n)

a) Given that 40% of the first application is parallelizable, how much speedup would you achieve with at application if run in isolation?

sol: Speedup = 1 / ((1-f)+f/n)
= 1 / ((1-0.4)+0.4/2)
= 1 / (0.6+0.2)
= 1 / 0.8
= 1.25

b) Given that 99% of the second application is parallelizable, how much speedup would this application observe if run in isolation?

sol: Speedup = 1 / ((1-f)+f/n)
= 1 / ((1-0.99)+0.99/2)
= 1 / (0.01+0.495)
= 1 / 0.505
= 1.98

c) Given that 40% of the first application is parallelizable, how much overall system speedup would you observe if you parallelized it?

sol: Speedup = 1 /(0.2+0.8((1-f)+f/n))
= 1 / (0.2+0.8*0.8)
= 1 / (0.2+0.64)
= 1 / 0.84
= 1.19

d) Given that 99% of the second application is parallelizable, how much overall system speedup would you get?

sol: Speedup = 1 /(0.8+0.2((1-f)+f/n))
= 1 / (0.8+0.2*0.505)
= 1 / (0.8+0.101)
= 1 / 0.901
= 1.1

e) If we follow fixed-time scalable up principle, and assume only parallelizable portion will scale up in size, what is the fixed-time speedup of question a) and b), respectively?

sol: For fixed-time scalable up principle, Speedup = (1-f)+nf
We need to calculate fixed-time speedup of question a) and b).

a) Speedup = (1-f)+nf
=((1-0.4)+0.4*2)
= 0.6 +0.8
= 1.4

b) Speedup = (1-f)+nf

=((1-0.99)+0.99*2)
= 0.01 +1.98
= 1.99

f) If we further assume that the applications are dense matrix computations, that is the memory requirement increases with n2 and computation increases with n3. Repeat e) for memory bounded speedup.

sol: For, dense matrix computations, that is the memory requirement increases with n2 and computation increases with n3.

Speedup = ((1-f)+f.pow(n,3/2)) / ((1-f)+f.pow(n,1/2))

a) Speedup = ((1-0.4)+0.4.pow(2,3/2)) / ((1-0.4)+0.4.pow(2,1/2))
= (0.6+0.4*2.8) /(0.6+0.4*1.4)
= 1.72/1.16
= 1.48

b) Speedup = ((1-0.99)+0.99.pow(2,3/2)) / ((1-0.99)+0.99.pow(2,1/2))
= (0.01+0.99*2.8) / (0.01+0.99*1.4)
= 2.78/ 1.39
= 2.78/1.39
= 2.007

2. Amdahl's law : Amdahl's law states that if a portion of a computation, can be improved by a factor n,and other portion cannot be improved, then the portion that cannot be improved will quickly dominate the performance, and further improvement of the improvable portion will have little effect.
Speedup is defined as sequential execution time over parallel execution time in parallel processing.

Let f be the portion of the work load that can be parallelized and n be the number of processors.
Then, the parallel processing speedup implied by Amdahl's law is:

Speedup = 1 / ((1-f)+f/n)

When n increased to infinity, speedup cannot be increased to infinity.

n=infinity, then, speedup (infinity)= 1/(1-f) ,since most applications have a sequential potion that cannot be parallelized, parallel processing is not scalable. Speedup (infinity) is referred to as a sequential bottle neck of multiprocessor systems.

3.a) What is the speedup we have obtained from fast mode?
SOl : we know that that, an enhancement to a computer that improves some mode of execution by a factor of 10.

Let us assume x= original execution time, and y= new execution time
x= 50% of y + 50% of y *10
= y* 50/100 + y*(50/100)*10
= 0.5y+5y
= 5.5y

Speedup = original execution time / new execution time
= x/y

= 5.5y/y
=5.5

The speedup we have obtained from fast mode is 5.5.

b) What percentage of the original execution time has been converted to fast mode?

SOl : we know that non-enhanced execution time is equals to enhanced mode execution time speedup by 10 in the new code.

=> (1-p) = p/10
11p=10
p=0.909

90.9% percentage of the original execution time has been converted to fast mode

4.a) Amdahl's law
sol: Speedup= n/(1+(n-1)a)
=20/(1+19*0.2)
= 20/4.8
= 4.166

b) Gustafson's law
sol: Speedup= a+(1-a)n
= 0.2 + (1-0.2)20
= 0.2 + 0.8*20
= 0.2+16
= 16.2

c) Sun/Ni's law, assuming G(20)=10
sol: Speedup= (a+(1-a)G(n))/ (a+(1-a)G(n)/n)
= (0.2 + (1-0.2)10) / (0.2+(1-0.2)10/20)
= 8.2 /0.6
= 13.66

7. Sol: Given, seven tasks with running times of 1, 2, 3, 4, 5, 5, and 10 units.
we need to compute the best- and worst-case speedup for a centralized scheme for dynamic mapping with two processes.

The time taken to complete all the tasks with single process = 30 units.
Best case speedup will occur when the processes share the load equally.
=> Process 1 completes : t1,t2,t3,t4,t5 tasks = 15 units
=> Process 2 completes : t6,t7 tasks = 15 units
It hapens when tasks assigned in the following order p1(t1), p2(t6),p1(t2), P1(t3), p2(t7), p1(t4),p1(t5) time taken to complete tasks = 15 units .

Therefore, Best case Speed Up = 30/15 = 2

Worst case speedup will occur when the processes share the load unequally, one processes takes more time and other process takes less time.

=> Process 1 completes: t1,t2,t3,t4,t7 tasks = 20 units
=> Process 2 completes: t5,t6 tasks = 10 units
It happens when tasks assigned in the following order p1(t1), p2(t5),p1(t2), P1(t3), p2(t6), p1(t4),p1(t7).

Therefore, worst case Speed Up = 30/20 = 1.5