CS546 “Parallel and Distributed Processing”
Homework 8

Submission:
Due by 11:59pm of 11/16/2016
Total points 100 - Late penalty: 10% penalty for each day late
Please upload your assignment on Blackboard with the following name:
CS546_SectionNumber_LastName_FirstName_HW8.
Please do NOT email your assignment to the instructor and/or TA!

1. (20 points) Your company has bought a new 8-core processor, and you have been asked to optimize your software for this processor. You will run two applications 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.
   a) Given that 40% of the first application is parallelizable, how much speedup would you achieve with at application if run in isolation?
   b) Given that 99% of the second application is parallelizable, how much speedup would this application observe if run in isolation?
   c) Given that 40% of the first application is parallelizable, how much overall system speedup would you observe if you parallelized it?
   d) Given that 99% of the second application is parallelizable, how much overall system speedupwould you get?
   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?
   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.

Ans:

a) Let serial Time be T_s and parallel Time be T_p
P=number of cores =8
Hence Speedup = $\frac{\mathrm{Ts}}{\mathrm{Tp}}$
Now as 40% of application can be parallelised. Hence
α=.6
Hence Speedup = $\frac{1}{\alpha+\frac{1-\alpha}{p}}$ =$\frac{1}{0.6+\frac{1-0.6}{8}}$=$\frac{1}{0.6+\frac{.4}{8}}$=1.538

b) Let serial Time be T_s and parallel Time be T_p
P=number of cores =8
Hence Speedup = $\frac{\mathrm{Ts}}{\mathrm{Tp}}$
Now as 99% of application can be parallelised. Hence
α=.01

Hence Speedup = $\frac{1}{\alpha+\frac{1-\alpha}{p}}$ =$\frac{1}{0.01+\frac{1-0.01}{8}}$=$\frac{1}{0.01+\frac{.99}{8}}$=$\frac{1}{0.13375}$=7.48

c) Now as Application uses only 80% of the resources. That means instead of p it will use .8p

Hence Speedup = $\frac{1}{\alpha+\frac{1-\alpha}{.8*p}}$ =$\frac{1}{0.6+\frac{1-0.6}{0.8*8}}$=$\frac{1}{0.6+\frac{.4}{6.4}}$=1.51

d) Now as Application uses only 20% of the resources. That means instead of p it will use .2p

Hence Speedup = $\frac{1}{\alpha+\frac{1-\alpha}{0.2*p}}$ =$\frac{1}{0.01+\frac{1-0.01}{0.2*8}}$=$\frac{1}{0.01+\frac{.99}{0.2*8}}$=1.59

e) Fixed Time scale up

$$\alpha_A = \frac{1}{1+\frac{(1-\alpha_G)*p}{\alpha_G}} \equiv \alpha_G = \frac{p*\alpha_A}{1+\alpha_A(p-1)}$$

1) $\alpha_G = \frac{8*.6}{1+.6(8-1)} = .923$

Speedup=$\alpha_G + (1-\alpha_G)p$=$0.923 + (1-0.923)8$=1.539

2) $\alpha_G = \frac{8*.01}{1+.01(8-1)} = 0.091592$

Speedup=$\alpha_G + (1-\alpha_G)p$=$0.091592 + (1-0.091592)8$=7.358856

f) Memory Bound scale up

1) Total Work=Computational work + Memory Constraint
= (n^3*(.4/p+.6)+ $n^{2*3/2}$)==n^3*(.65+1)

Speedup=$\frac{\alpha+(1-\alpha)W}{\alpha+\frac{(1-\alpha)W}{p}}$=$\frac{.6+(1-.6)1.65n^3}{.6+\frac{(1-.6)1.65n^3}{8}}$=$\frac{.6+.01n^3}{.6+\frac{.01n^3}{8}}$ $\frac{.6+.01n^3}{.6+0.00125n^3}$

2) Total Work=Computational work + Memory Constraint
= (n^3*(.99/p+.01)+ $n^{2*3/2}$)==n^3*(1.13375)

Speedup=$\frac{\alpha+(1-\alpha)W}{\alpha+\frac{(1-\alpha)W}{p}}$=$\frac{.01+(1-.01)1.13375n^3}{.01+\frac{(1-.01)1.13375n^3}{8}}$=$\frac{.01+1.12n^3}{.01+\frac{1.12n^3}{8}}$ $\frac{.01+1.12n^3}{.01+0.14n^3}$

2. (15 points) Prove Amdahl's law (i.e., when p goes to infinite, under Amdahl's law the speedup is ..?).
Ans:

The execution time of the whole task before the improvement of the resources of the system is denoted *T*. It includes the execution time of the part that does not benefit from the improvement of the resources and the execution time of the one that benefits from it. The percentage of the execution time of serial program is denoted α. The one concerning the part that is parallel is therefore 1 − α. Then

$$T = \alpha T + (1-\alpha)T$$

It is the execution of the part that benefits from the improvement of the resources that is speed up by the factor *p* after the improvement of the resources. Consequently, the execution time of the part that does not benefit from it remains the same, while the part that benefits from it becomes

$$\frac{1-\alpha}{p}T$$

T$_p$ = $\frac{1-\alpha}{p}T + (\alpha)T$

Speedup for a fixed Workload W is

S=$\frac{TW}{T_pW}$=$\frac{T}{T_p}$=$\frac{T}{\frac{1-\alpha}{p}T+(\alpha)T}$=$\frac{1}{\frac{1-\alpha}{p}+(\alpha)}$

When $p \to \infty$

S=$\frac{1}{(\alpha)}$

Hence as per this the speedup is inversely proportional to percentage of serial code.

3. (10 points) Assume that we make an enhancement to a computer that improves some mode of execution by a factor of 10. Enhanced mode is used 50% of the time, measured as percentage of the execution time when the enhanced mode is in use. Recall that Amdahl's Law depends on the fractions of the original, non-enhanced execution time that could make use of enhanced mode.$_p$ Thus, we cannot directly use this 50% measurement to compute speedup with Amdahl's Law.
   a) What is the speedup we have obtained from fast mode?
   b) What percentage of the original execution time has been converted to fast mode?

Ans:

Let Time taken without enhancement is T

As enhancement is applied for 50% of the time Hence For that time it will improve performance

Ie $T_P = \frac{1}{2}\left(\frac{T}{10}\right) + \frac{1}{2}T = \frac{11}{20}T$

a) Speedup= $\frac{20}{11} = 1.82$

b) Percentage of Time converted to fastMode=$\frac{T_F}{T} * 100 = \frac{\frac{T}{20}}{T} * 100$ =5%

4. (15 points) Calculate the speedup of the three models of parallel speedup assuming the fraction that cannot be parallelized is a = 0.2, and the number processors is p = 20.
   a) Amdahl's law
   b) Gustafson's law
   c) Sun/Ni's law, assuming G(10)=5

Ans:

a) S$_A$=$\frac{1}{\alpha+\frac{1-\alpha}{p}} = \frac{1}{.2+\frac{1-.2}{20}} = \frac{1}{.2+\frac{.8}{20}} = 4.16$

b) $\alpha_A = \frac{1}{1+\frac{(1-\alpha_G)*p}{\alpha_G}} \equiv \alpha_G = \frac{p*\alpha_A}{1+\alpha_A(p-1)}$=$\frac{20*.2}{1+.2(20-1)} = .83$

S$_G$=$\alpha_G + (1-\alpha_G)p = 0.83 + (1-0.83) * 20$=4.23

c) W=5

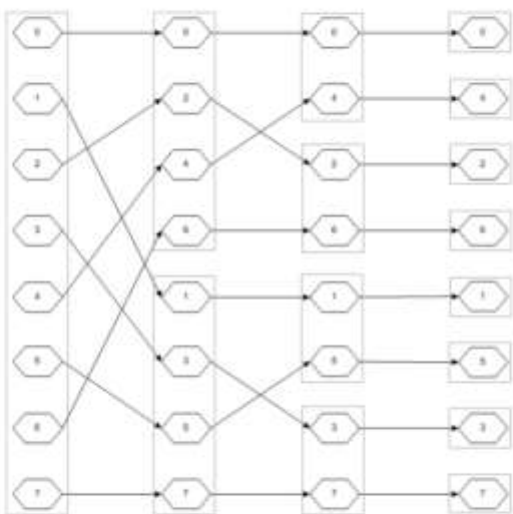
S$_M$=$\frac{\alpha+(1-\alpha)W}{\alpha+\frac{(1-\alpha)W}{p}}$=$\frac{.2+(1-.2)5}{.2+\frac{(1-.2)5}{20}} = \frac{.2+(.8)5}{.2+\frac{(.2)5}{5}} = \frac{42}{4} = 10.5$

5. (20 points) In the algorithm shown, assume a decomposition such that each execution of Line 7 is a task. Draw a task-dependency graph and a task-interaction graph.

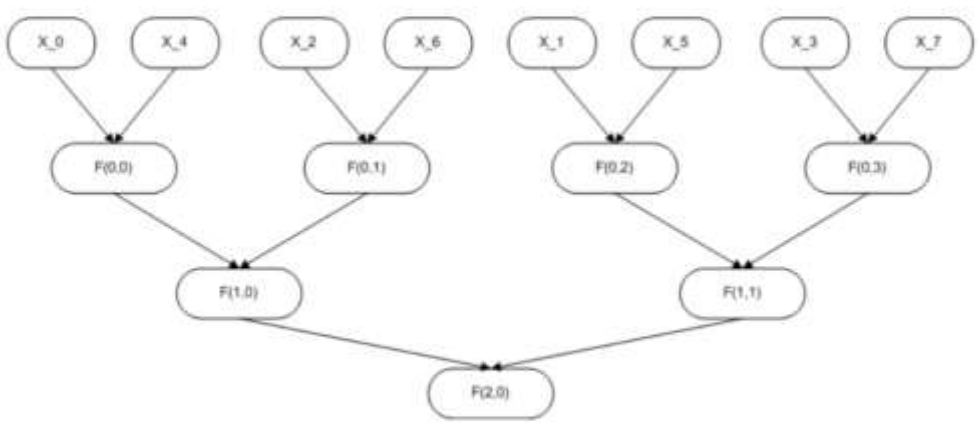
```
1.    procedure FFT_like_pattern(A, n)
2.    begin
3.       m := log2 n;
4.       for j := 0 to m - 1 do
5.            k := 2^j;
6.            for i := 0 to n - 1 do
7.                 A[i] := A[i] + A[i XOR 2^j];
8.       endfor
9.    end FFT_like_pattern
```

Ans : for n=8

Task dependency graph



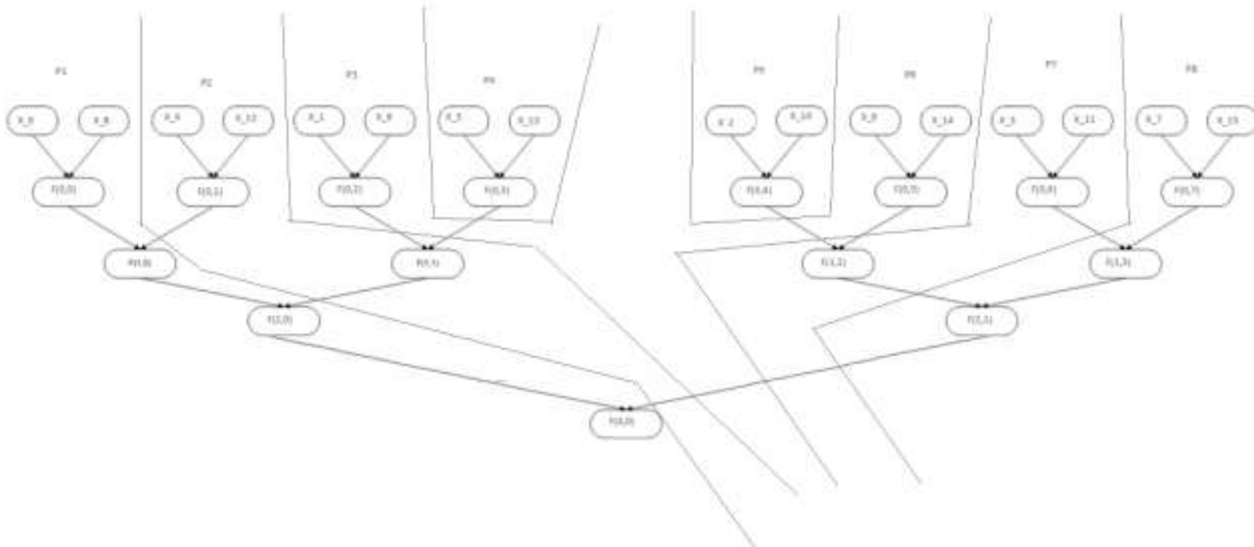
Task interaction graph



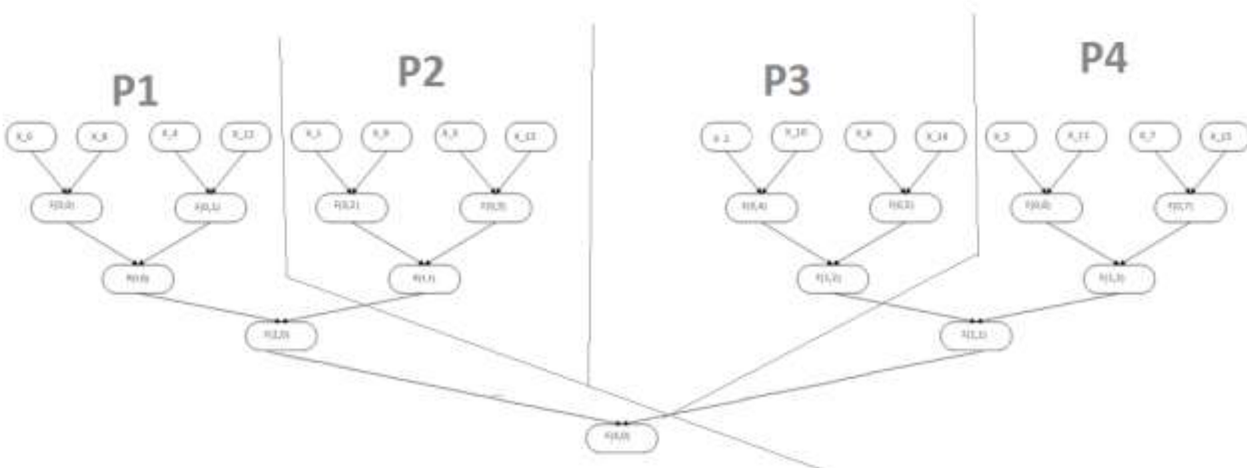
6. (10 points) In the above algorithm, if n = 16, devise a good mapping for:
   a) 16 processes
   b) 8 processes

Ans

For 16 processors



For 8 processors



7. (10 points) Consider seven tasks with running times of 1, 2, 3, 4, 5, 5, and 10 units, respectively. Assuming assigning work to a process does not take any time, compute the best- and worst-case speedup for a centralized scheme for dynamic mapping with two processes.

Ans:
Serial unit of time = 30
Worst Case

T	M	P1	P2
1	1,2	1	2
1	3	3	2
2	4	3	4
2	5	5	4
3	5	5	5
2	10	10	5
8	10	10	

Tp=19
S=1.51

Best case

T	M	P1	P2
5	10,5	10	5
5	5	10	5
3	4,3	4	3
1	2	4	2
1	1	1	2

Tp=15
S=2

Note: We encourage collaboration between you and your classmates. Discuss various approaches and techniques to better understand the questions. However, we do NOT allow copying solutions or code. This is considered as cheating and falls under IIT code of honor. Penalties will be enforced. Please make sure you write your own solutions.
GOOD LUCK!